FINAL PROGRAM

TRISTAN 2019: Schedule for Monday 17 June

8:00	Registration Opens		
8:45	**Welcome**		
9:00	**Plenary: David Simchi-Levi -- Online Resource Allocation with Applications to Revenue Management**		
10:00	Morning Tea		
	Endeavour Room - Resort Side	**Endeavour Room - Beach Side**	**Chart Room**
	Chair: Alexander Paz	***Chair: Michael Forbes***	***Chair: Teo Crainic***
10:30	A Gaussian Process Approach for High-dimensional Simulation-based Transportation Optimization. Timothy Tay and Carolina Osorio	On the needs for on-demand management of ridesharing mobility. Andrea Simonetto, Julien Monteil and Claudio Gambella	Locomotive Fuel Management with Inline Refueling. Ahmad Kazemi, Andreas Ernst, Mohan Krishnamoorthy and Pierre Le Bodic
11:00	Continuous simulation optimization of expensive black-box traffic systems: A review of algorithms and applications to toll pricing. Ziyuan Gu, Meead Saberi and S Travis Waller	A Decentralized Shared CAV System Design & Application. Seyed Mehdi Meshkani, Shadi Djavadian and Bilal Farooq	Finding robust shunting plans. Roel van den Broek, Han Hoogeveen and Marjan Van Den Akker
11:30	Incorporating competition in demand-based optimization models. Stefano Bortolomiol, Virginie Lurkin and Michel Bierlaire	Passenger-centric dial-a-ride problem for on-demand mobility systems. Shadi Sharif Azadeh, Yousef Maknoon, Bilge Atasoy, Michel Bierlaire and Moshe Ben Akiva	Railway Rolling Stock Maintenance Scheduling. Lukas Bach and Daniel Palhazi Cuervo
12:00	Specification of Mixed Logit Models Assisted by an Optimization Approach. Alexander Paz and Cristian Arteaga	Applying Fragments to the Dial a Ride Problem. Michael Forbes	Intermodal Rail Blocking and Car Fleet Management. Teodor Gabriel Crainic, Emma Frejinger and Tien Mai
12:30	Lunch		
	Chair: Ali Haghani	***Chair: Konstantinos Zografos***	***Chair: Mahboobeh Moghaddam***
13:30	On the use of operations research methods for the design of school districts. Karen Smilowitz	A Continuous Model for Electric Vehicle Sharing with Battery Degradation. Jian Wu, Xin Wang and Feng Ju	A new Benders decomposition method for metropolitan container logistics problems. Andrew Perrykkad, Andreas Ernst and Mohan Krishnamoorthy
14:00	Solving The Joint Multi-School Bell Time and Route Scheduling Optimization Problem. Ali Haghani, Ali Shafahi and Zhongxiang Wang	A mathematical model and a solution algorithm for the electric vehicle routing problem with non-stationary battery swapping. Ramin Raeesi and Konstantinos G Zografos	Forecasting a freight carrier's demand for container shipments. Greta Laage, Emma Frejinger and Gilles Savard
14:30	Optimizing the Training Transfer of Junior Soccer Players. Christian Jost, Alexander Döge, Sebastian Schiffels and Rainer Kolisch	Combinatorial Auction with Bidder-Defined Items for Fractional Ownership of Autonomous Vehicles. Mahdi Takalloo, Aigerim Bogyrbayeva, Hadi Charkhgard and Changhyun Kwon	Improving Drayage Operations through a Realistic Optimization Model. Mahboobeh Moghaddam, Robin H Pearce, Hamid Mokhtar and Carlo Prato
15:00	Afternoon Tea		
	Chair: Mohsen Ramezani	***Chair: Simon Dunstall***	***Chair: Natashia Boland***
15:30	Incentive-Compatible Mechanisms for Traffic Intersection Auctions with Autonomous Vehicles. David Rey, Vinayak Dixit and Michael Levin	Integrated robust & possibilistic multiobjective humanitarian logistic model with social costs. Cristián E Cortés, Pablo A Rey and Luis E Yáñez	Sending a reliable cost-efficient flow through a stochastic time-varying network. Alberto Giudici, Tao Lu, Clemens Thielen and Rob Zuidwijk
16:00	Analytical Delay Models for Interrupted Mixed Flow of Autonomous and Human-Driven Vehicles. Reza Mohajerpoor and Mohsen Ramezani	Fleet sizing and operations management in wildfire suppression operations. Simon Dunstall, Nicholas Davey, Carolyn Huston, Edmundo Claro-Rodriguez and Saman Halgamuge	Pickup and delivery problem with truckload synchronization through multiple cross-docks. Yousef Maknoon and Gilbert Laporte
16:30	Max-Pressure Based Autonomous Intersection Management with Pedestrians. Rongsheng Chen, Jeffrey Hu, Michael Levin and David Rey		Integer Programming Models for Freight Logistics Service Network Design with In-Tree Constraints. Natashia Boland and Ira Wheaton

TRISTAN 2019: Schedule for Tuesday 18 June

	Endeavour Room - Resort Side	Endeavour Room - Beach Side	Chart Room
	Chair: Rajan Batta	***Chair: Satish Ukkusuri***	***Chair: Tarun Rambha***
8:30	Recent Advancements in Solution Methods for Traveling Salesman Problems with a Drone. Mark Bierema, Eveline van Dijck and Paul Bouman	An approach to model competition in ridesharing. Venktesh Pandey, Julien Monteil and Andrea Simonetto	On the Price of Satisficing in Network User Equilibria. Mahdi Takalloo and Changhyun Kwon
9:00	Time-Dependent Vehicle Routing Problem with Time Windows on a Road Network. Maha Gmira, Michel Gendreau, Andrea Lodi and Jean-Yves Potvin	Modeling the Operation Dynamics of Ride-sourcing Markets. Xinwu Qian, Rui Chen, Chao Yang and Satish Ukkusuri	Traffic-dependent limited unfairness in a system optimum traffic assignment. M Grazia Speranza, Enrico Angelelli and Valentina Morandi
9:30	A mixed integer programming approach for scheduling spatially distributed jobs with degradation rate: application to pothole repair. Rajan Batta and Fatemeh Aarabi	Matching Passengers and Drivers with Multiple Objectives in Ride Sharing Markets. Guodong Lyu, Wangchi Cheung, Chung Piaw Teo and Hai Wang	Identifying Compliant Users Needed for Social Optimum Routing in Traffic Networks. Tarun Rambha, Michael Albert, Guni Sharon, Stephen Boyles and Peter Stone
10:00	Morning Tea		
10:30	**Plenary: SC Wong -- Continuum Modeling Approach to Land Use, Transport and the Environment for Urban Cities**		
	Chair: Rob Zuidwijk	***Chair: Ludovic Leclerq***	***Chair: Marco Rinaldi***
11:30	Spatial and temporal synchronization of truck platoons. Anirudh Kishore Bhoopalam, Niels Agatz and Rob Zuidwijk	Network performance under different levels of ride-sharing: A simulation study. Negin Alisoltani Dehkordi, Ludovic Leclercq and Mahdi Zargayouna	Investigating the robustness of route-based sensor location policies under variable network demand. Marco Rinaldi and Francesco Viti
12:00	Truck Platooning Network Design. Szymon Albinski, Teodor Gabriel Crainic and Stefan Minner	A Dynamic Ride-Sourcing System for City-Scale Networks. Amir Hosein Valadkhani and Mohsen Ramezani	Connected Vehicle Sensor Location Model for Traffic Congestion Mitigation. Hyoshin Park and Ali Haghani
12:30	Lunch		
	Chair: Pirmin Fontaine	***Chair: Nigel Wilson***	***Chair: William Lam***
13:30	Trade-offs in shared transportation services. Margaretha Gansterer, Richard F Hartl and Sarah Wieser	The co-development of railway and land use in Sydney. Bahman Lahoorpoor and David Levinson	Dynamic traffic assignment for multimodal GSOM models. Megan Khoshyaran and Jean-Patrick Lebacque
14:00	The Vehicle Routing Problem with Digital Lockers Terminals. Simona Mancini	Passenger-to-Itinerary Assignment Model for Congested Urban Rail Networks. Yiwen Zhu, Haris Koutsopoulos and Nigel Wilson	An equilibrium service choice in a dynamic traffic assignment with real-time information. Nam H Hoang, Hai L Vu and Dong Ngoduy
14:30	The Vehicle Routing Problem with Load-Dependent Travel Times for Cargo Bike Routing. Pirmin Fontaine	Contributions of demand variability to unreliability in the public transport system. Emily Moylan	The Complementary Duet of Vehicular Diverging: An Experimental Approach. Mingyue Sheng and Siwen Pan
15:00		Analytical BusPlus. Arthur Mahéo and Michael Forbes	An activity-based approach for optimizing the High-Occupancy Toll lanes in congested road networks. Dang Khoa Vo and William H K Lam
15:30	Afternoon Tea		
	Chair: Martin Savelsbergh	***Chair: Joseph Chow***	***Chair: Ma ëlle Zimmermann***
16:00	A Branch-and-Cut-and-Price Algorithm for the Capacitated Location-Routing Problem. Pedro Henrique P V Liguori, A Ridha Mahjoub, Ruslan Sadykov and Eduardo Uchoa	Shared Autonomous Mobility Fleets and Multimodal Transit Networks: Design Methodology and Trade-Offs. Hani Mahmassani, Helen Pinto and Michael Hyland	Applying Meta-heuristic Algorithm with parallel computation framework to simulation-based Dynamic Traffic Assignment. Mostafa Ameli, Jean-Patrick Lebacque and Ludovic Leclercq
16:30	Optimizing Package Express Operations in China. Baris Yildiz and Martin Savelsbergh	A many-to-many stable matching cost allocation model for multimodal Mobility-as-a-Service. Saeid Rasulkhani, Theodoros Pantelidis and Joseph Chow	A strategic Markovian equilibrium model for capacitated networks. Maëlle Zimmermann, Emma Frejinger and Patrice Marcotte
17:00		Data-Driven Transit Network Design at Scale. Dimitris Bertsimas, Yee Sian Ng and Julia Yan	

19:00	**Meeting of International Scientific Committee [Placeholder]**

TRISTAN 2019: Schedule for Thursday 20 June

	Endeavour Room - Resort Side	Endeavour Room - Beach Side	Chart Room
	Chair: Vikrant Vaze	***Chair: Michiel Bliemer***	***Chair: Anton Kleywegt***
8:30	A Passenger-Centric Approach to Air Traffic Flow Management. Alexandre Jacquillat	General Solution Scheme for the Static Link Transmission Model. Mark Raadsen and Michiel Bliemer	Dynamic Flexible Time Window Pricing for Attended Home Deliveries. Charlotte Köhler, Jan Fabian Ehmke, Ann Campbell and Catherine Cleophas
9:00	Integrated airline schedule, aircraft and passenger recovery: incorporating passenger response to disruptions. Luis Cadarso and Vikrant Vaze	Stable Primal Numerical Method for the Bottleneck Model. Hillel Bar-Gera	Pricing for Drivers and Customers for Goods Deliveries. Luce Brotcorne, Anton Kleywegt and Youcef Magnouche
9:30	Choice-Based Integrated Airline Fleet Assignment and Schedule Design. Chiwei Yan, Cynthia Barnhart and Vikrant Vaze	Dynamic speed control and lane management in the general link transmission model. Michiel Bliemer, Mark Raadsen, Luc Wismans and Luuk Brederode	Decision-Based Scenario Clustering for Decision-Making Under Uncertainty: applications in transport planning. Michael Hewitt, Janosch Ortmann and Walter Rei
10:00	Morning Tea		
10:30	**Plenary: Stephane Hess -- Quantum Logic and Neural Preference Accumulation: A Leap Into the Unknown or a New Dawn for Dynamic Travel Behaviour Models?**		
	Chair: Alexandre Jacquillat	***Chair: Nicholas Molyneaux***	***Chair: David Levinson***
11:30	A Large-scale Neighborhood Search Approach to Airport Slot Allocation. Nuno Antunes Ribeiro, Alexandre Jacquillat and António Pais Antunes	Improving pedestrian dynamics by preventing counter-flow.. Nicholas Molyneaux, Riccardo Scarinci and Michel Bierlaire	A General Theory of Access. David Levinson
12:00	Optimizing multi-level, multi-objective airport slot-scheduling decisions. Fotios Katsigiannis and Konstantinos G Zografos	Toward Development of a Link Transmission Model for Pedestrian Networks. Tanapon Lilasathapornkit, Wei Liu and Meead Saberi	Estimating Travellers' Trip Purposes using Public Transport Data and Land Use Information. Bo Du
12:30	Lunch		
	Chair: Mehmet Yildirimoglu	***Chair: Dong Ngoduy***	***Chair: Benoit Montreuil***
13:30	Multi-reservoir MFD-based simulation: An application to the city network of Lyon. Guilhem Mariotte, Mahendra Paipuri and Ludovic Leclercq	The Full Cost of Auto Accessibility. Mengying Cui and David Levinson	Optimizing Omni-Channel Fulfilment with Store Transfers. Joydeep Paul, Niels Agatz and Martin Savelsbergh
14:00	Perimeter Flow Control with Time-Varying Cordon based on Macroscopic Fundamental Diagram. Ye Li, Reza Mohajerpoor and Mohsen Ramezani	A predictive model of lane-changing possibilities: deep learning approach. Seunghyeon Lee and Ngoduy Dong	Reliable Parcel Routing Policy in a Physical Internet. Ido Orenstein and Tal Raviv
14:30	Finding critical links to estimate a Macroscopic Fundamental Diagram in congested urban networks. Elham Saffari, Mehmet Yildirimoglu and Mark Hickman	A Study on Driver's Stopping Behavior Focusing on Generalization. Hirotoshi Shirayanagi, Takahiro Tsubota, Shinya Kurauchi and Toshio Yoshii	Operations Design for High-velocity Intra-city Package Service. Iman Dayarian, Adolfo Rocco Rocco, Alexander Stroh, Martin Savelsbergh, Alejandro Toriello and Alan Erera
15:00	Design of urban transportation infrastructure for optimal passenger throughput. Allister Loder, Michiel C J Bliemer and Kay W Axhausen	A Novel Traffic Estimation Approach Using Multi-Source Data on Motorways. Xuan Sy Trinh, Dong Ngoduy, Mehdi Keyvan-Ekbatani and Blair Robertson	Hyperconnected Urban Parcel Logistic Systems Design. Benoit Montreuil, Sara Kaboudvand, Louis Faugere and Martin Savelsbergh
15:30	Afternoon Tea		
	Chair: Francois Soumis	***Chair: Michael Bell***	***Chair: Frederic Semet***
16:00	Integrated Scheduling and Flow Management in Air Traffic Management Networks. Kai Wang and Alexandre Jacquillat	Simultaneous correction of the time and location bias associated with a reported crash by exploiting the spatiotemporal evolution of travel speed. Zhengli Wang and Hai Jiang	Estimating Vehicle Fleet Composition for Last-Mile Delivery Service. Frederic Semet, Ekaterina Alekseeva, Luce Brotcorne, Youcef Magnouche and Etienne Soufflet
16:30	Machine Learning feeding Mathematical Programming for Air Crew Scheduling. Francois Soumis, Yassine Yaakoubi and Simon Lacoste-Julien	The impact of road capacity on connectivity by eigenvector centrality analysis. Hiroe Ando, Michael Bell and Fumitaka Kurauchi	Multi-Period Workload Balance in Last-Mile Urban Delivery. Yang Wang, Lei Zhao and Martin Savelsbergh
17:00	A MIP formulation for the flexible rostering of ground personnel at an international airport. Juan Pablo Cavada, Cristián E Cortés and Pablo A Rey		
19:00	**Conference Dinner at Auditorium**		

TRISTAN 2019: Schedule for Friday 21 June

9:00	**Plenary: Michael Bell -- Designing Greener City Logistics Networks**		
10:00	Morning Tea		
	Endeavour Room - Resort Side	**Endeavour Room - Beach Side**	**Chart Room**
	Chair: Alan Erera	***Chair: Kosuke Kawakami***	***Chair: Andreas Ernst***
10:30	A Priori Routing for Strategic Time Slot Management in Online Grocery Retailing. Thomas Visser and Martin Savelsbergh	A column generation procedure for the Flexible Ship Loading Problem. Jonas Christensen and Dario Pacino	A Survivable p-Hub Median Problem and a Modied Benders Decomposition Method. Hamid Mokhtar, Mohan Krishnamoorthy and Andreas T Ernst
11:00	The pickup and delivery problem with on-line transfers. Paul Bouman, Gizem Ozbaygin and Lucas Veelenturf	Ship routing problem with berthing time clash avoidance constraints and minimizing demurrage. Kosuke Kawakami and Mirai Tanaka	Stochastic Single-Allocation Hub Location. Nicolas Kämmerling, Borzou Rostami, Christoph Buchheim, Joe Naoum-Sawaya and Uwe Clausen
11:30	Tactical Design of Same-Day Delivery Systems. Alexander Stroh, Alan Erera and Alejandro Toriello		An Intermodal Hub Location Problem for Container Distribution in Indonesia. Hamid Mokhtar, Perwira Redi, Mohan Krishnamoorthy and Andreas T Ernst
12:00	**Closing and Box (Takeaway) Lunch**		

Multi-Output Gaussian Process in Simulation-based Transportation Optimization

Timothy Tay
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
Email: tayt@mit.edu

Carolina Osorio
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

1 Introduction

In transportation optimization problems, deriving an accurate analytical form of the objective function is often difficult. This is due to the nonlinearity of driver choices, as well as interactions between vehicles on the road. As a result, this problem is often circumvented using traffic simulators. These simulators make use of various driver behavior models to replicate the real-life driving behavior of drivers on the road, while also keeping track of many quantities of interest of every vehicle in the network to intricate detail [1], such as travel time, fuel consumption, number of stops, etc. The level of detail provided by the traffic simulators have thus made them popular tools for obtaining stochastic estimates of the performance of predetermined transportation strategies.

However, the high level of detail in traffic simulators comes with a challenge – they tend to be computationally expensive to evaluate. This is particularly so when dealing with a large network involving many vehicles. Furthermore, in a simulation-based optimization (SO) problem, large networks can also mean more variables to optimize, which leads to an exponential increase in computational demand [2]. Hence, given the computational cost of running a simulation, an optimization framework that obtains a good solution with fewer simulation evaluations is desirable.

The SO problems we consider here have simulation-based, continuous and general (e.g. non-convex) objective functions, with unknown analytical form. The constraints are assumed analytical and differentiable. In such cases, it is crucial for the optimization algorithm to balance exploration and exploitation in order to find a good solution. Exploration involves sampling points to improve knowledge of the feasible region, while exploitation involves using that knowledge to identify better

solutions [3]. Bayesian optimization is a global optimization framework which tries to balance the exploration-exploitation tradeoff [4], and has become a popular approach recently for a wide range of problems, including those in the field of transportation [5]. It has been shown that Bayesian optimization techniques remain efficient in terms of the number of simulation evaluations needed.

Our past work in SO has focused on the formulation of metamodels, which combine information from analytical models and the simulator, to approximate the objective function [6, 7]. However, these past approaches do not explicitly try to balance exploration and exploitation. For instance, a general-purpose sampling strategy (e.g., uniform random sampling) is often used for exploration. There is potential to improve the performance of SO algorithms by exploiting the structural information of analytical network models to design suitable exploration-exploitation techniques.

To address these issues, we formulate a methodology that combines ideas from multi-output Gaussian process (GP), Bayesian optimization and analytical network modeling [6, 7]. Bayesian optimization provides a way to balance exploration and exploitation when selecting the next point to evaluate. At the same time, the multi-output GP accounts for the inaccuracies in the analytical model, which are reflected in the prediction uncertainty of the simulator estimate. Also, since the multi-output GP exploits structural information of the analytical model through its correlation with the simulator, this means that even if the model does not directly estimate the objective function, but instead provides an estimate of another correlated measure, the model can still assist the optimization. This opens up the possibility of using a wide range of analytical models for optimizing a given objective function. For example, two models that are accurate only in different regions of the search space could theoretically supplement one another in predicting the simulator output. Such a framework for optimization can be applied to a variety of transportation problems.

The exploitation of structural information of the analytical model can be explained by Figure 1 [8]. In Figure 1a, function 3 is to be minimized. It is anti-correlated (resp. correlated) to function 1 (resp. function 2). If observations from functions 1 and 2 are not used in the GP predictions for function 3, the prediction uncertainty in regions without observations would be large. This is shown

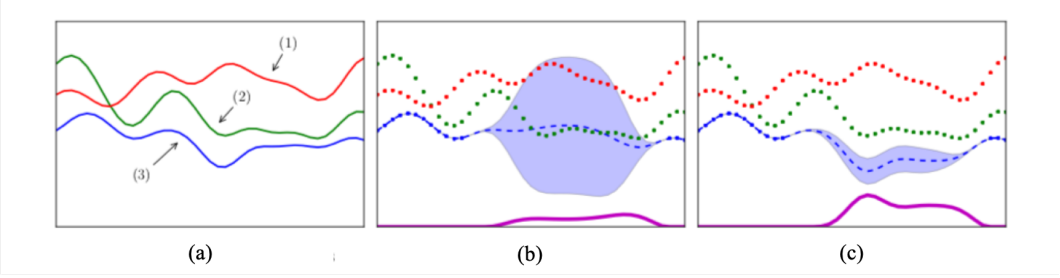


Figure 1: 1-dimensional illustration of how multi-output works. (a) Given three correlated functions, where function 3 is to be minimized, (b) independent GP predictions lead to greater uncertainty for function 3 in regions without observations, whereas (c) multi-output GP predictions rely on observations from the other two functions to reduce prediction uncertainty for function 3. Figure adapted from [8].

in Figure 1b. Observations are denoted by dots, and the GP prediction mean and uncertainty for function 3 are represented by the dashed line and shaded area respectively. The line at the bottom of the plot denotes the acquisition function (more details in Section 2.3), where it is shown that the maximum does not correspond to the minimum of the actual function. On the other hand, using a multi-output GP, observations from functions 1 and 2 can help with the predictions in the region without observation for function 3. This is illustrated in Figure 1c, where the prediction uncertainty (shaded area) is smaller, and the prediction mean (dashed line) better represents the actual function. The maximum of the acquisition function is also close to the actual minimum.

In the following section, we formulate the problem and proposed method in more detail. In Section 3, we discuss the empirical set up used to validate the proposed approach.

2 Methodology

2.1 Problem Formulation

Transportation SO problems can generally be formulated as:

$$\min_{\mathbf{x}\in\chi} f(\mathbf{x}, z; p) \equiv \mathbb{E}[F(\mathbf{x}, z; p)] \tag{1}$$

where f is the objective function, F represents the stochastic output of a simulation run, $\mathbf{x}$ is the vector of decision variables, χ is the feasible region, z denotes the endogenous variables and p represents the deterministic exogenous parameters.

The proposed methodology is relevant for a variety of continuous SO problems. Here we illustrate its use with a large-scale network-wide traffic signal control. The objective function f can be taken to be the expected trip travel time, and the decision vector $\mathbf{x} = (x_1, \ldots, x_d)$ consists of the green splits (i.e. normalized green times) for each signal phase. Then, z would account for route choice decisions, departure times, etc.; p would account for external traffic demand, traffic network topology, cycle times, offset, etc.

2.2 Multi-Output Gaussian Processes

Single-output GPs are a class of models that attempt to approximate a function with a scalar output ($f : \chi \to \mathbb{R}$). GPs are completely specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. Given a set of N previously evaluated points $X = [\mathbf{x_1}, \ldots, \mathbf{x_N}]^T$ and their function estimates $\mathbf{y} = (y_1, \ldots, y_N)$, the GP estimate at a test point $\mathbf{x}_*$ is normally distributed with a predictive mean $\mu(\mathbf{x}_*)$ and variance $\sigma(\mathbf{x}_*)$ given by:

$$\mu(\mathbf{x}_*) = \mathbf{k}^T[\mathbf{K} + \sigma_n^2 I]^{-1}(\mathbf{y} - m(\mathbf{x}_*)) \tag{2}$$

$$\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k} \tag{3}$$

$$\mathbf{k} = [k(\mathbf{x}_*, \mathbf{x_1}), \ldots, k(\mathbf{x}_*, \mathbf{x_N})] \tag{4}$$

and $\mathbf{K}$ is the covariance matrix for all pairs of training points in the set X. Here, we assumed that the function outputs $\mathbf{y}$ contain an additive i.i.d. Gaussian noise with variance σ_n^2 for the n^{th} training point. We refer the reader to [9] for more details on single-output GP regression.

Multi-output GPs extends the above framework to functions with vector-valued outputs ($\mathbf{f} : \chi \rightarrow \mathbb{R}^T$). It can also model correlated functions, by using the correlation to reduce uncertainty on the estimates. This is done by defining the multi-output covariance function $k((\mathbf{x}, t), (\mathbf{x}', t'))$ to denote the similarity between outputs $f_t(\mathbf{x})$ and $f_{t'}(\mathbf{x}')$. Given $k((\mathbf{x}, t), (\mathbf{x}', t'))$, the standard GP framework can be applied to obtain the predicted means and variances for the various outputs.

2.3 Acquisition Function

In the Bayesian optimization framework, given the updated GP model, the acquisition function serves as a means to guide the search for the optimum. The acquisition function is usually defined in a way that tries to balance exploration and exploitation. It has a high value in regions where the predicted mean is small (in a minimization problem), and also in regions where the prediction uncertainty is large and there is significant probability that the optimum could lie in those regions. The point to evaluate by simulation is then chosen by maximizing the acquisition function.

The expected improvement (EI) criterion [4] is a popular choice of acquisition function, and it has been shown to be effective in many studies. Hence, we will use it in this study. It is defined as

$$EI(\mathbf{x}) = (y_{best} - \mu(\mathbf{x}))\Phi(Z) + \sigma(\mathbf{x})\phi(Z) \tag{5}$$

$$Z = \frac{y_{best} - \mu(\mathbf{x})}{\sigma(\mathbf{x})} \tag{6}$$

where y_{best} is the simulator output of the current best point; $\phi(\cdot)$ and $\Phi(\cdot)$ denote the pdf and cdf of the standard normal distribution respectively.

3 Empirical Analysis

We first evaluate the performance of the Bayesian optimization framework using multi-output GPs on a simple toy network, consisting of just 4 controlled intersections and 8 signal phases. It is used to show that multi-output GP is able to leverage the analytical model to find a good solution with fewer simulation evaluations, compared to using an independent GP.

Furthermore, we consider a case-study using a simulation model of Midtown Manhattan. The Midtown Manhattan model is a large-scale network, and constitutes a high-dimensional optimization problem in the area of simulation optimization. It consists of 97 controlled intersections with a total of 259 signal phases. Using this simulation model, we compare the performance of Bayesian optimization with other SO methods, in terms of the quality of proposed solutions, as well as the number of simulation evaluations required to obtain those solutions.

References

[1] A. Pell, A. Meingast and O. Schauer, "Trends in Real-time Traffic Simulation", *Transportation Research Procedia* 25, 1477-1484 (2017).

[2] S. Shan and G.G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions", *Structural and Multidisciplinary Optimization* 41, 219-241 (2010).

[3] R. Sutton and A. Barto, "Chapter 2: Multi-arm Bandits", in *Reinforcement Learning: An Introduction*, 31-52, MIT Press, Cambridge, MA (1998).

[4] D. Jones, M. Schonlau, W. Welch, "Efficient global optimization of expensive black-box functions", *Journal of Global Optimization* 13, 455-492 (1998).

[5] X.M. Chen, L. Zhang, X. He, C. Xiong, Z. Li, "Surrogate-based optimization of expensive-to-evaluate objective for optimal highway toll charges in transportation network", *Computer-Aided Civil and Infrastructure Engineering* 29, 359-381 (2014).

[6] C. Osorio, M. Bierlaire, "An analytic finite capacity queueing network model capturing the propagation of congestion and blocking", *European Journal of Operational Research* 196, 996-1007 (2009).

[7] C. Osorio, L. Chong, "A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems", *Transportation Science* 49, 623-636 (2015).

[8] K. Swersky, J. Snoek, R.P. Adams, "Multi-task Bayesian Optimization", *Advances in Neural Information Processing Systems*, 2004-2012 (2013).

[9] C.E. Rasmussen, C.K.I. Williams, "Chapter 2: Regression", *Gaussian Processes for Machine Learning*, 7-31, MIT Press, Cambridge, MA (2006).

On the needs for on-demand management of ridesharing mobility

Andrea Simonetto
IBM Research, Ireland Lab

Julien Monteil
IBM Research, Ireland Lab
julien.monteil@ie.ibm.com

Claudio Gambella
IBM Research, Ireland Lab

1 Introduction

Urban mobility has witnessed significant changes in recent years via the rise of smart mobility services such as dynamic ridesharing [1]. This unprecedented growth in the mobility offering raise concerns as it brings more vehicles on the roads, hence more congestion and more environmental pollution. City authorities are well aware of this problem and start to introduce legislation to control this growth, such as capping the number of licences awarded to ride-hailing vehicles [2]. Such measures seem intuitively of good will but it remains to be seen how they affect the level of performance, such as the service rate for the customers. In this work, we first present our solution to solve city-scale ridesharing and then present an analysis on how the fleet vehicles, if allocated in a static way, may lead to decreasing the quality of service.

We provide a solution for solving the centralized real-time city-scale ridesharing problem, with similarities but differences with the recent work of [3], by mapping incoming batch of requests with available vehicles, in a four-step procedure: (i) selecting candidate vehicles to serve requests, (ii) computing serving costs meeting ridesharing constraints by solving a DARP problem, given those computed costs (iii) performing optimal assignments of requests to vehicles, and (iv) performing rebalancing of vehicles, if needed. In particular one contribution of our work is that we highlight that linear assignments can perform equally as good as more elaborated assignments, when run at a high enough sampling rate. This is particularly interesting as it enables the possibility of solving linear assignment problems involving multiple companies in a privacy-aware fashion, see e.g. [4]. Then, we extend the analysis to a multi-company scenario where each company has a given number of operating vehicles.

2 City-scale ridesharing solution

Given a set $\mathcal{M}$ of m customer trip requests at time t and a ridesharing fleet $\mathcal{C} = \{1, \dots, n\}$ of vehicles, each with capacity C_i, we are interested in determining ridesharing solutions in real-time. Our ridesharing service aim at providing an assignment of requests (customers) to available vehicles and their correspondent routes, according to some optimization criteria. Available vehicles are those that can pick up customers, while complying with the time constraints associated with the requests, and without exceeding their seat capacity.

We express the ridesharing service as an optimization engine, which is run at specific time periods t_k $(k = 0, 1, 2, \dots)$. At each such time instant t_k, the service processes the requests submitted by customers in the time window $(t_{k-1}, t_k]$, and find optimal vehicle-costumer assignment and correspondent routes. The trip requests include the origin and destination points, the desired time of departure, the maximum waiting time, detour time and journey time. Our algorithm leverages the following feature: at each optimization run, at most one new request (customer) is assigned to a vehicle. This design choice enables us to reduce the ridesharing optimization problem into a sequence of linear assignment problems, which can be solved very efficiently.

Hence, the ridesharing logic is run at every sampling period t_k and works as follows:

1. It obtains the customer requests submitted during the time interval $[t_{k-1}, t_k)$.
2. It passes requests to the context mapping module that returns at most 2max$n$ vehicles per request (max$n$ being a scalar).
3. It asks insertions costs to the vehicles returned by the context mapping module (DARP).
4. It runs the linear assignment problem with the updated costs.
5. It sends to customers and vehicles the assignments and their corresponding routes.
6. If some customers cannot be serviced, it calls an internal rebalancing module, which runs the logic again from (2. to 5.) with loosen time constraints and for idle vehicles only.

We test our approach on two datasets: the New York City dataset [5] and the Metropolitan Melbourne dataset [6] (instance M1). Some results are highlighted on Table 1 and 2. $C(\Sigma)$ denotes the cost function used in [3], h denotes the sampling rate (time interval), SR the percentage of requests satisfied, y and n the waiting and detour time in the presence and absence of a rebalancing component. Table 1 reports comparable results with the literature while much lower computational times. Table 2 shows interesting results in the context of a much wider area and for scheduled (not instantaneous) requests.

Table 1: NYC results for the entire demand and comparison with [3] (indicated by *).

vehicles	c	max$n$	cost	h [s]	SR [%]	waiting y [min]	waiting n [min]	detour y [min]	detour n [min]	comp. time [s]
2000	4	25	TD	10	92.10	3.95	3.88	3.41	3.40	10.10
3000	4	25	TD	10	99.87	3.31	3.23	2.60	2.59	7.87
*, 2000	4	-	$C(\Sigma)$	30	93.70	-	3.28	-	3.29	57.55
*, 3000	4	-	$C(\Sigma)$	30	97.91	-	2.70	-	2.28	51.55

Table 2: Results for the Melbourne Metropolitan Area dataset.

vehicles	customers [%]	c	max$n$	cost	h [min]	SR [%]	detour y [min]	detour n [min]	comp. time [min]
600	100.0	4	10	TD	2	75.68	5.90	5.91	1.99
800	100.0	4	10	TD	2	96.06	5.58	5.58	2.61
1000	100.0	4	10	TD	2	100.00	4.87	4.87	2.96

3 On regulating numbers of operating vehicles

In the New York City setting, we consider the case of e.g. 2 companies, sharing the whole customer set in Manhattan. In such a context of competition, we are facing this problem: the more cars the companies have on the roads, the better service they can offer to the users, and the more market share they will get, as they guarantee better quality of service, and gain reputation. Hence there is a need to limit the number of operating vehicles. A first step in this direction, albeit primitive, is to cap the number of ridesharing licenses, in the light of the recent article [2].

Therefore, we investigate the situation in which the central agent enforces a total number of licenses per company, and as a result allocates the customers to the companies based not only on a global welfare cost (our cost function TD) *but* also on a defined market share, for instance proportional to the number of issued licenses for the companies.

In this scenario, the two companies have defined market share (say 25% and 75%) and the city authority enforces it via carefully engineering the cost function, so that the two companies expect to receive 25% and 75% of the city customers, respectively. As an example, we consider the scenario at a given time instant t_k of company 1 having 100 operating vehicles with 75% of market share, while company 2 having 50 operating vehicles and with 25% of market share.

Table 3: Comparison between optimal and market share solutions; market shares: 75% and 25%.

vehicles	c	max$n$	method	h [s]	SR [%]	waiting y [min]	waiting n [min]	detour y [min]	detour n [min]
150	4	8	TD	10	95.75	3.54	3.43	2.53	2.53
100+50	4	16	TD'	10	89.49	4.83	4.72	4.76	4.76

Table 3 reports the performance of the scenario with respect to a centralised setting -one company only. As we see, performance deteriorates on all the metrics (even if we augment max$n$ to 16). In order to obtain a similar level of performance to the monopolistic case, company 1 will have to increase its fleet size, because company 2 will most likely not reduce its fleet, unless enforced to. As a result, the total fleet size will increase, as well as the number of vehicles on the roads and traffic congestion. Consequently, the engineering of the cost function in a static way to ensure a fair distribution of the users across the companies is not a good one. Instead, the city authorities must look into adapting the operating fleet size of the companies to the time-varying demand, i.e. number of requests per time interval $[t_{k-1}, t_k)$. This underlines the needs for strong policies by city authorities to regulate the number of cars utilized by ridesharing companies, in a real-time fashion.

References

[1] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.

[2] New York Times. Uber Hit With Cap as New York City Takes Lead in Crackdown. `https://www.nytimes.com/2018/08/08/nyregion/uber-vote-city-council-cap.html`, 8 August 2018. Online; 8 August 2018.

[3] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[4] Oshri Naparstek and Amir Leshem. Fully distributed optimal channel assignment for open spectrum access. *IEEE Transactions on Signal Processing*, 62(2):283–294, 2014.

[5] NYC Taxi and Limousine Commission dataset. available online.

[6] Ride-sharing instances for the Melbourne Metropolitan Area. available online at: https://github.com/davidrey123/Ridesharing.

Locomotive Fuel Management with Inline Refueling

Ahmad Kazemi, Andreas Ernst
School of Mathematical Sciences
Monash University, Clayton, Victoria, Australia
Email: ahmad.kazemi@monash.edu

Mohan Krishnamoorthy
School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia

Pierre Le Bodic
Faculty of Information Technology
Monash University, Caulfield, Victoria, Australia

1 Introduction

Fuel and fuel-related expenses constitute a major part of the railroad companies operating costs. They are the second largest operational cost source of the railroad companies in the US [1] and the third largest in Australia [2]. Therefore, even small improvements in fuel management of railroad companies may contribute to annual savings of millions of dollars [3].

Locomotive fuel management usually consists of three interrelated problems [4]. The first problem arises due to the fact that fuel prices vary at different locations. Therefore, fuel plans might be optimized in a manner that the locomotives be refueled as much as possible at the "cheap" stations, possibly more than the amount that it is required to reach next station, to reduce the need for refueling at the "expensive" stations. This strategy is called *tankering*. Moreover, each refueling operation incurs a fixed cost. Therefore, the first two problems are scheduling the refueling operations of each locomotive and determining the amount of each refueling operation. By considering only these two problems, the fuel management of a fleet of locomotives can be decomposed to each locomotive and be solved separately as a variant of the Lot Sizing Problem. Railroad companies have to pay a contract fee to use a fuel station during the time horizon. Hence, location decisions should be also taken into account. Considering each of these problems separately generally results in sub-optimal solutions. The integerated problem of these three sub-problems first introduced and solved by a Lagrangian framework by Nourbakhsh and Ouyang [3].

The majority of the literature on this problem is motivated by an INFORMS Problem Solving Competition [4, 5, 6].

In networks that have long distances between stations relative to the locomotives' tank capacity, the tankering strategy is limited since the tanks capacity may not allow many successive trips. Railroad companies employ inline refueling, operated by inline tanks, to overcome this problem. An inline tank is a large fuel reservoir which connects to the locomotives and can refuel them during the trip while they are moving. Since the inline tanks are a substantial investment, the number of available inline tanks is limited and thus they should be managed efficiently. Although inline tanks are already in use in the railroad industry, to the best of our knowledge, there is no mathematical approach to plan the inline refueling. In this paper, we introduce the problem of locomotive fuel management with inline refueling (LFMIR) and discuss the employed mathematical modeling approach and the problem complexities.

2 Problem Statement

Given a schedule of the trains and the locomotives over a specified time horizon and different fuel prices at different stations, LFMIR consists in determining the fuel stations to use, the schedules and amounts of refueling operations of the locomotives and inline tanks, and fuel plans of the locomotives and the inline tanks as well as the assignment of the inline tanks to the locomotives and the amount of inline refueling operations. Locomotives must not run out of fuel during their journeys between two stations. We also consider a variant of LFMIR, called "*iLFMIR*", in which determining the location of using fuel stations and scheduling the refueling operations are ignored to keep only binary variables that are related to inline tanks.

3 Methodology

The locomotive fuel management without inline tanks has been proved to be NP-hard [5]. LFMIR is also NP-hard since it is a generalization of the problem without inline refueling. Moreover, *i*LFMIR is also NP-hard. This is interesting, as the iLFMIR is simply a Min-Cost Network Flow problem to which inline tanks have been added.

Corollary 3.1 *LFMIR is NP-hard.*

Theorem 3.1 *iLFMIR, obtained by ignoring the decisions on the location of using fuel stations and the scheduling refueling operations from LFMIR, is NP-hard.*

We propose a Mixed-Integer Program based on the representation of the problem on a time-space network to solve the LFMIR. In the proposed time-space network, each arc and each node

represent a different activity, such as refueling, making a trip, and inline refueling. Although the network components correspond to different activities, from the modeling point of view, the arcs transfer the fuel flow, and the nodes conserve and distribute the fuel flow. The decisions on the location of stations, scheduling refueling operations, and the assignment of the inline tanks correspond to the decision of using an arc. Therefore, the proposed MIP consists of three parts, namely, a Network Design problem, a Network Flow problem, and the linkage of these problems.

4 Results

The proposed model is applied on a real-world case in Australia and the INFORMS Problem Solving Competition case [4]. The MIP is implemented in Python using CPLEX 12.8.0. The results are discussed in two directions; the economic impacts of using the proposed models and the computational demonstrations of the complexity of the problem.

4.1 Economic Impacts

The results on the Australian and the INFORMS cases suggest significant cost-savings in comparison with the current practice of business and also the literature optimization models. For the Australian case, using the MIP for iLFMIR results in about \$110,000 weekly savings in comparison with the current practice, which is remarkable with respect to the required investment on the inline tanks. Moreover, the proposed models improve the company's operation in terms of dealing with robustness. The INFORMS instances are too complex to be solved by our approach for more than three inline tanks. Therefore, we divide this case into smaller tractable instances. The results on the instances show about \$32,000 fortnightly savings on average in comparison with the literature optimization models which do not consider inline refueling.

4.2 Complexity Numerical Analyses

In addition to proving the NP-hardness of the problem, we show that the resulting MIP model is significantly harder to solve by CPLEX when there are inline tanks available. Table 1 shows the results by CPLEX in 12 hours of computing time. For LFMIR, as the number of inline tanks increases from 3, the cost of the best integer solution increases due to the significant optimality gaps. Moreover, even with 1 inline tank, the optimality gap is more than one percent which is considerable from the cost point of view. As Table 1, the same phenomenon occurs even if we solve iLFMIR as the inline tank fleet size increases from 5, which shows the growth of complexity due to the inclusion of inline refueling. The same results are also observed for the Australian case. Experiments with different LP solution methods demonstrate that it is very time consuming just to solve the root relaxation of the problem.

Table 1: Computational results for 12 hour runs on an instance of the INFORMS data

Fleet Size	*i*LFMIR		LFMIR	
	Total Costs ($)	Optimality gap (%)	Total Costs ($)	Optimality gap (%)
0	10,863,126	0.00	11,399,804	0.04
1	10,794,739	0.01	11,307,349	1.18
2	10,744,647	0.03	11,248,985	1.50
3	10,703,411	0.05	11,251,773	2.06
4	10,672,793	0.07	11,415,804	4.00
5	10,659,298	0.19	11,494,898	4.99
10	11,318,609	6.66	12,161,406	100.00
15	11,318,609	7.01	11,777,626	100.00
20	11,318,609	7.18	11,782,376	100.00

5 Conclusions

In this paper, we introduced a new class of the fuel management problems of a locomotive fleet. We also prove that LFMIR and its relaxed version are NP-hard. We employed a time-space network structure to develop a Mixed-Integer Program to tackle this problem. Based on the case studies, we presented results in two directions. First, the economic impacts of the proposed models are investigated to demonstrate the potential benefits. Second, our experiments show the problem is too complicated for current off-the-shelf solvers. For future research, we intend to propose specialized solution algorithms for the problem. Inline refueling follows some specific rules. Therefore, the other research avenue is to incorporate these rules to mimic the inline refueling behavior. Finally, incorporating strategic decisions on inline tanks fleet size and establishing new stations is another research direction. Strategic decisions are for the longer time horizons in which the problem parameters cannot be assumed deterministic. Therefore, incorporating uncertainty is required to consider the strategic decisions.

References

[1] Union Pacific Corporation, "Annual Report 2017", Union Pacific Corp. Nebraska (2018).

[2] Asciano, "2016 Annual Report", Asacanio Limited Victoria (2016).

[3] S.M. Nourbakhsh and Y. Ouyang, "Optimal fueling strategies for locomotive fleets in railroad networks", *Transportation Research Part B: Methodological* 44, 1104-1114 (2010).

[4] Problem Solving Competition, "Railway Applications Section (RAS)", INFORMS Hanover, MD (2010).

[5] T. Raviv and M. Kapsi, "The locomotive fleet fueling problem", *Operations Research Letters* 40, 39-45 (2012).

[6] V.P. Kumar and M. Bierlaire, "Optimizing Fueling Decisions for Locomotives in Railroad Networks", *Transportation Science* 49, 149-159 (2015).

Continuous simulation optimization of expensive black-box traffic systems: A review of algorithms and applications to toll pricing

Ziyuan Gu*

Department of Civil and Environmental Engineering

University of New South Wales

Email: ziyuan.gu@unsw.edu.au

Meead Saberi

Department of Civil and Environmental Engineering

University of New South Wales

S. Travis Waller

Department of Civil and Environmental Engineering

University of New South Wales

1 Introduction

Simulation optimization (SO) refers to, without loss of generality, the minimization of an objective function subject to a set of constraints, both of which are evaluated through computer simulations. It is an active area of research particularly in transportation engineering where simulation-based dynamic traffic assignment (SDTA) models have been widely developed and used for solving various network design problems (NDPs). See [1] for a recently developed SDTA model of Melbourne, Australia as well as a state-of-the-art survey.

SO per se is not a new concept, see [2] for a recent decent overview. Using SO to solve transportation NDPs is not new either, see [3, 4]. The major concern, however, is the computational efficiency associated with an SO method since the formulation of a typical NDP in a large-scale SDTA environment is often characterized by a computationally expensive objective function, a high-dimensional decision vector comprising multiple decision variables, and simulation noise [5]. Also, given the complex formulation, there are likely multiple local minima from which an SO method needs to escape in order to locate the global optimum, or, more generally, one of the global optima if more than one exist.

Therefore, this study aims to consider and compare different computationally efficient global SO methods for solving expensive transportation NDPs. Specifically, we apply different SO methods to a developed benchmark toll level problem (TLP) and investigate their comparative performance.

2 SO methods

In this study, we are interested in continuous NDPs and hence discrete SO methods are not considered. To the best of our knowledge, different continuous SO methods to date can be classified into seven broad categories: (i) random search or metaheuristics, (ii) response surface method (RSM), (iii) stochastic approximation (SA), (iv) direct search, (v) estimation of distribution algorithms (EDAs), (vi) Lipschitzian optimization, and (vii) feedback control. Given an expensive TLP to be solved in this study, random search and EDAs are left out because of their demanding requirement of a large number of function evaluations. Direct search as a local optimizer is not considered either as we are interested in global optimizers.

We consider and compare the most representative and perhaps the best performing SO method for each of the four identified categories: (i) regressing kriging (RK) for RSM [5], (ii) simultaneous perturbation SA (SPSA) for SA [6], (iii) DIRECT (DIviding RECTangles) for Lipschitzian optimization, and (iv) proportional-integral (PI) controller for feedback control [7]. To reduce simulation noise of a stochastic traffic simulator commonly rendered by different random seed numbers, we can couple standard fixed- or "smarter" variable-number sample path optimization [8] with the above methods. However, computer simulations often display what one might call numerical noise as well, i.e., the objective function evaluations tend to scatter about a smooth trend rather than lying on it [9]. Due to the space constraint, we only present a summary of the four SO methods without full mathematical details.

Table 1 Summary of the four SO methods

Method	Mechanism	Capabilities		Overheads
		Objective	Constraint	
RK	Approximating the simulation input-output mapping by a limited number of sample points and a mathematical construct	Any	Any	Parameter estimation and infill point sampling
SPSA	Using finite-difference approximation to enable gradient descent	Any	Any	Parameter tuning
DIRECT	Diving the parameter space into (hyper)rectangles based on sample point evaluations	Any	Any	Potentially optimal (hyper)rectangle identification
PI	A trial-and-error method to gradually reduce the error from the set point	Set point	Bound	Parameter tuning

3 Benchmark TLP

Consider the following benchmark TLP:

$$\min_{\boldsymbol{\tau}\in\Omega} \mathrm{E}\left[\sum_{h=1}^{m} |\bar{K}_h - K_{\mathrm{cr}}|\right] \tag{1}$$

s.t.

$$\bar{K}_h = DTA(\boldsymbol{\tau}), \qquad h = 1,2,\dots,m \tag{2}$$

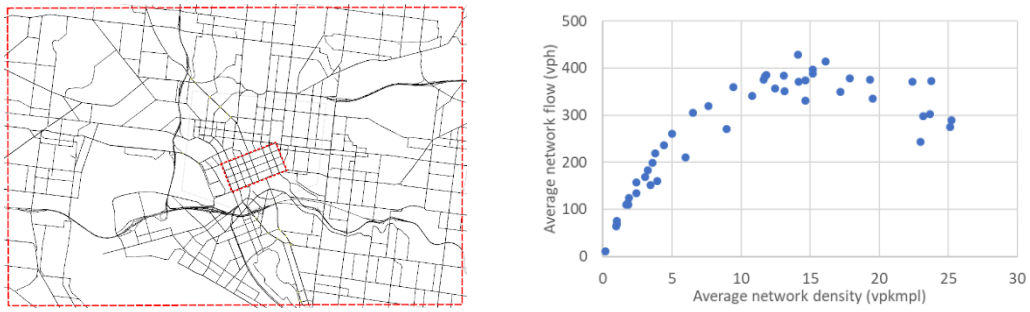
$$\Omega = \{\boldsymbol{\tau} | \boldsymbol{\tau}_{\min} \le \boldsymbol{\tau} \le \boldsymbol{\tau}_{\max}\} \tag{3}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_m]^{\mathrm{T}}$ is the toll rate decision vector for the m tolling intervals, $\bar{K}_h$ is the average network density within the h-th tolling interval, K_{cr} is the critical network density identified from the

network fundamental diagram (NFD), $\mathrm{E}[\cdot]$ is the expectation operator, $DTA(\cdot)$ is the black-box function of the simulation model, and Ω is the feasible set of toll rates with $\boldsymbol{\tau}_{\min}$ being the lower bound and $\boldsymbol{\tau}_{\max}$ being the upper bound. The objective function aims to minimize the expected summation of the absolute difference between $\overline{K}_h$ and K_{cr} for the m tolling intervals. As such, the network is pricing-controlled near the critical network density without entering the congested regime of the NFD, thereby achieving the maximum network productivity [10].

4 Preliminary results

Figure 1(a) illustrates the SDTA model used in this study while Figure 1(b) shows the simulated NFD of the pricing zone (PZ) over the 6-10 AM peak period. K_{cr} is hence set at 15 vpkmpl and the tolling period is set between 8 and 9 AM with two 30-min tolling intervals, i.e. $m = 2$. The pricing regime investigated is a linear distance toll, i.e., the toll price is linearly proportional to the distance traveled within the PZ. To accelerate and simplify the comparison, we only use a same random seed number.



(a) (b)

Figure 1 (a) The extracted sub-network from the greater Melbourne area model where the inner rectangle represents the PZ, and (b) simulated NFD of the PZ without pricing

Figure 2(a) shows the simulated NFD of the PZ after applying the optimal toll rates obtained from the PI method. Convergence is achieved with only around 10 function evaluations as shown in Figure 2(b). There is, as expected, a large hysteresis loop in the NFD, which is fully discussed in [7]. Figure 2(c) illustrates, on one hand, the search path of the PI method, and, more importantly, the numerical noise - there is a wide range of function values in the relatively small global optimal region. Figure 2(d) shows the contour plot for the DIRECT method with over 500 function evaluations while Figure 2(e) shows the constructed response surface by the RK method with far fewer 50 function evaluations. Although both methods successfully pinpoint the global optimum, the DIRECT method is much more computationally expensive probably due to the presence of high numerical noise - see the highly irregular and non-smooth contours. Figure 2(f) displays the search path of the SPSA method which successfully steps into the global optimal region within 50 function evaluations but somehow jumps out of it and moves away. We are currently performing further numerical tests to finalize our findings.

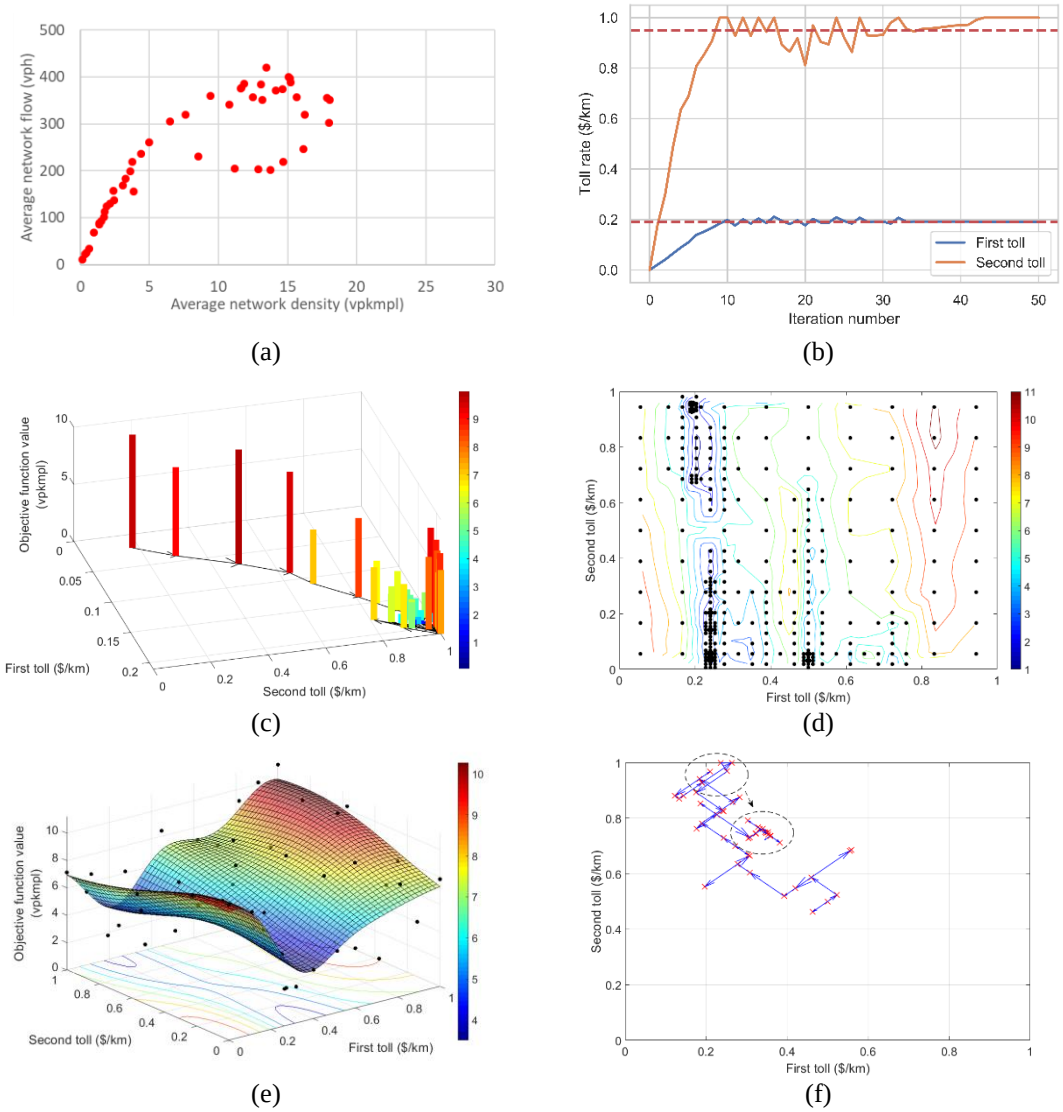


Figure 2 (a) Simulated NFD of the PZ after applying the optimal toll rates obtained from the PI method, (b) convergence of the PI method, (c) numerical noise along the search path of the PI method, (d) contour plot for the DIRECT method, (e) constructed response surface by the RK method, and (f) search path of the SPSA method

5 Conclusion

Four computationally efficient SO methods are investigated and compared in this study on a developed benchmark TLP. Results so far suggest that the PI method is most suited for solving simple problems with a set point objective and bound constraints, while the RK method is the best performing solution to more complex problems.

References

[1] S. Shafiei, Z. Gu, and M. Saberi, "Calibration and Validation of a Simulation-based Dynamic Traffic Assignment Model for a Large-Scale Congested Network," *Simulation Modelling Practice and Theory*, vol. 86, pp. 169-186, 2018.

[2] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Simulation optimization: a review of algorithms and applications," *Annals of Operations Research*, vol. 240, no. 1, pp. 351-380, 2016.
[3] C. Osorio and M. Bierlaire, "A simulation-based optimization framework for urban transportation problems," *Operations Research*, vol. 61, no. 6, pp. 1333-1345, 2013.
[4] C. Osorio and L. Chong, "A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems," *Transportation Science*, vol. 49, no. 3, pp. 623-636, 2015.
[5] X. Chen, L. Zhang, X. He, C. Xiong, and Z. Li, "Surrogate-Based Optimization of Expensive-to-Evaluate Objective for Optimal Highway Toll Charges in Transportation Network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 5, pp. 359-381, 2014.
[6] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332-341, 1992.
[7] Z. Gu, S. Shafiei, Z. Liu, and M. Saberi, "Optimal distance- and time-dependent area-based pricing with the Network Fundamental Diagram," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 1-28, 2018.
[8] G. Deng and M. C. Ferris, "Variable-Number Sample-Path Optimization," *Mathematical Programming*, vol. 117, no. 1-2, pp. 81-109, 2009.
[9] A. I. J. Forrester, A. J. Keane, and N. W. Bressloff, "Design and analysis of "noisy" computer experiments," *AIAA Journal*, vol. 44, no. 10, pp. 2331-2339, 2006.
[10] C. F. Daganzo, "Urban gridlock: Macroscopic modeling and mitigation approaches," *Transportation Research Part B: Methodological*, vol. 41, no. 1, pp. 49-62, 2007.

A Decentralized Shared CAV System Design & Application

S. Mehdi Meshkani*, Shadi Djavadian, Bilal Farooq
Ryerson University, Toronto, Canada
Email: smeshkani*, shadi.djavadian, bilal.farooq @ryerson.ca

1 Introduction

Shared connected and autonomous vehicles (SCAVs) are an emerging mobility mode which combines the lower price of public transit with flexibility and availability of on-demand vehicles. SCAVs are capable of deploying dynamic ride-sharing strategy to service multiple traveler trips simultaneously resulting in reduction in the fleet size, vehicle miles traveled, and operating costs [1]. Most of the research efforts related to SCAVs have considered a centralized dispatching system in their proposed frameworks [1, 2]. High computational complexity as a result of the large amount of information that need to be processed in real-time is one of the main issues related to centralized dispatching [3]. To overcome the above-mentioned shortcoming, a large-scale decentralized SCAV dispatching system is proposed through a simulation model. To evaluate the designed system, it is implemented on an in-house agent based simulation platform already developed by Djavadian and Farooq [3]. In this platform, an end to end distributed dynamic vehicle routing for CAVs (E2ECAV) using a network of intelligent intersections (I^2) was proposed. In their proposed platform I^2s track the dynamic state of the network and guide CAVs in the network. We extend this platform for a SCAV system by developing a decentralized dispatcher utilizing I^2s.

2 Methodology and Results

This study is divided into two distinctive parts. In the first part, to evaluate the effectiveness of the proposed decentralized SCAVs dispatching system, a centralized SCAV dispatching system as a benchmark is developed. Both systems have the capability of ride-sharing and are evaluated on the network of downtown Toronto. The second part, on the other hand, focuses on enhancing ride-matching problem mentioned in the first part using a mathematical optimization model.

2.1 Network Structure

The structure of simulation environment consists of two network layers and three types of agents. The two network layers are: communication network and the physical road network. Physical road network is represented by a network $G(I, L)$, which consists of I intersections (nodes/vertices) and L links (edges). Agents in this system are vehicle agents ($v \in V$), passenger agents ($p \in$P), and infrastructure agents. There are two types of vehicle agents, including normal traffic (M) and SCAV fleet ($f \in$F); Two types of infrastructure agents are available namely: link agents ($l \in$L) and intelligent intersection agents ($I_n^2 \in I^2$). There are two types of communications in this system, ($V2I$) and ($I2I$). Δ is dispatch update cycle which is set to 1 min in this study.

2.2 Decentralized and Centralized SCAV System

In the decentralized SCAV system, intersections aside from routing CAVs and SCAVs, play the role of distributed dispatchers with which each having a depot. Passengers request to the nearest intersection via smartphone. If there is any available SCAV at intersection depot, it is assigned to the passenger. If not, passenger waits for any passing SCAV with enough capacity. The general policy for dropping off is FIFO. Each SCAV after dropping off the last traveller goes back to its own depot. In the centralized version, there is a centralized controller which is responsible for receiving requests from passengers, matching them to SCAVs and ensuring that all travellers are served. In this system, there are just two depots, based on the feasible locations in the network.

2.2.1 Case Study and Results

The proposed algorithm was tested on a highly congested downtown Toronto network. The demand used here is based on the adjusted 2011 travel survey of Toronto to accommodate the 2018 growth factor. Two types of scenarios under different SCAV demand levels and fleet size for both centralized and decentralized were conducted. As can be seen in Figure 1 and Figure 2, average waiting time for decentralized SCAV system under various demand level and fleet size yields much better results than centralized one. The boxplot in Figure 3 shows the average waiting time for decentralized system. With the increase in fleet size, standard deviation gets smaller and closer to the average which means that almost all passengers experience a reasonable waiting time.

2.3 Ride-Matching with Decomposition Algorithm

The second part of this study focuses on improving the ride-matching process of the designed decentralized SCAV system. To do so, a binary optimization problem is formulated to minimize the total system-wide passengers' travel time in the network. In order to avoid computational complexity of the system-wide optimization problems, a decomposition algorithm is utilized to

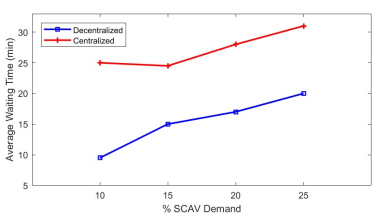


Figure 1: Wait Time vs Demand.

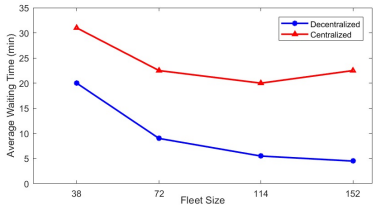


Figure 2: Wait Time vs Fleet Size.

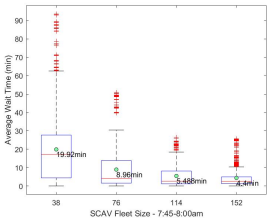


Figure 3: Wait Time variation

solve the original ride-matching problem by solving multiple sub-problems. These sub-problems are defined for intelligent intersections that are the smallest part of the network and play the role of dispatchers in the decentralized SCAV system. In the sub-problems, each intersection covers its neighborhood. For any new trip request the optimization problem is solved and the best SCAV from the intersection neighborhood is assigned to the trip request such that it minimizes passenger's travel time. By solving each sub-problem and summing over the whole intersections at each update dispatch time interval, the original ride-matching problem would be optimized.

Let I^2 be the set of intelligent intersections, in which each intersection is denoted by I_n^2, P be the set of passengers making request to the closest intersection I_n^2, and F be the set of SCAV fleet in the neighborhood of intersection. Any new passenger is denoted by $i \in P$, a SCAV in the neighborhood of intersection is represented by $j \in F$. K_j is the set of current passengers in SCAV j and current passenger travelling with SCAV j is denoted by $k_j \in K_j$. The main decision variable is x_{ij} which is equal to 1 if passenger i is matched with SCAV j, and 0 otherwise. $T_o^{k_j}$ is the original (without ride-sharing) in-vehicle time of current passenger k in SCAV j, $T_s^{k_j}$ is shared (with ride-sharing) in-vehicle time of current passenger k in SCAV j, D^{k_j} is detour time of current passenger k in SCAV j, $R_o^{k_j}$ is original remaining time of current passenger k in SCAV j, $R_s^{k_j}$ is shared remaining time of current passenger k in SCAV j, T_o^{ij} is original in-vehicle time of new passenger i if matched with SCAV j, T_s^{ij} is shared in-vehicle time of new passenger i if matched with SCAV j, t_i is passenger's allowed travel time, D^{ij} is detour time of new passenger i if matched with SCAV j, W_{ij} is the wait time of new passenger i if matched with SCAV j, c_j is current capacity of SCAV j. Passengers have time window such that e_i is the earliest departure time of new passenger i and l_i is his latest arrival time. Total number of intersections is represented by N and α, β, γ are parameters. Eq (1) is the system-wide objective function which minimizes passengers' total travel time over all network intersections. Eq (2) is defined for each intersection which aims at minimizing the total travel time of all passengers making request to the intersection.

$$\min Z = \sum_{n=1}^{N} I_n^2 \tag{1}$$

$$\min I_n^2 = \sum_i \sum_j (W_{ij} + T_s^{ij}) x_{ij} \qquad \forall I_n^2 \in I^2 \tag{2}$$

Eq (3) minimizes each new passenger's total travel time making request to the closest intersection which consists of wait time and in-vehicle time.

$$\min G = (W_{ij} + T_s^{ij})x_{ij} \tag{3}$$

subject to

$$D^{k_j} = T_s^{k_j} - T_o^{k_j} \qquad \forall k \in K_j, j \in F \tag{4}$$

$$D^{ij} = T_s^{ij} - T_o^{ij} \qquad \forall j \in F \tag{5}$$

$$t_i = l_i - e_i \tag{6}$$

$$D^{k_j} \leq \alpha T_o^{k_j} \qquad \forall k \in K, j \in F \tag{7}$$

$$R_s^{k_j} - R_o^{k_j} \leq \beta R_o^{k_j} \qquad \forall k \in K, j \in F \tag{8}$$

$$D^{ij} \leq \gamma T_o^{ij} \qquad \forall j \in F \tag{9}$$

$$W_{ij} + T_s^{ij} \leq t_i \qquad \forall j \in F \tag{10}$$

$$W_{ij} \leq 5 \text{ min} \qquad \forall j \in F \tag{11}$$

$$\sum_j X_{ij} \leq 1 \tag{12}$$

$$c_j \geq 1 \qquad \forall j \in F \tag{13}$$

$$x_{ij} = \{0, 1\} \qquad \forall j \in F \tag{14}$$

Eq. (4-6) define constrains representing current on-board passengers' detour time, new passenger's detour time, and new passenger's allowed travel time, respectively. Constraint (7-9) are adapted and modified based on Fagnant & Kockelman [1]. Constraint (7-8) ensure that detour time and remaining time increase of current on-board passenger are no more than a percent of their original in-vehicle and remaining time. Constraint (9) confirms that new passenger's detour time is no more than a percentage of their original in-vehicle time. Constraint (10) ensures that new passenger's total travel time is no more than his allowed travel time. Constraint (11) puts a threshold for passenger's waiting time. Constraint (12) ensures that every passenger is matched with no more than one car. Constraint (13) confirms that SCAV has enough capacity and (14) states the binary decision variables. We consider both event based and discrete time based optimization to evaluate which one and under what conditions works better.

References

[1] D.J. Fagnant and K.M. Kockelman, "Dynamic ride-sharing and optimal fleet sizing for a system of shared autonomous vehicles", *Transportation* 45 (1), 143–158 (2018).

[2] J.Ma, X. Li, F. Zhou, and W.Hao, "Designing optimal AV sharing and reservation systems", *Transp. Research Part C.* 84, 124–141 (2017).

[3] S. Djavadian and B. Farooq, "Distributed Dynamic Routing Using Network of Intelligent Intersections", In ITS Canada ACGM, Niagara Falls, April 2018.

Finding robust shunting plans

Roel van den Broek
Department of Information and Computing Sciences
Utrecht University
Email: r.w.vandenbroek@uu.nl

Han Hoogeveen
Department of Information and Computing Sciences
Utrecht University

Marjan van den Akker
Department of Information and Computing Sciences
Utrecht University

1 Introduction

The *Netherlands Railways (NS)*, the largest Dutch passenger railway operator, uses only a subset of the available trains during off-peak hours to operate the timetable. The surplus of rolling stock is parked at shunting yards, where the trains can be cleaned and maintained. To ensure that the shunting yards are operating efficiently, a feasible *shunting plan* needs to be created, which describes which activities, such as coupling and decoupling train units, service tasks and train movements, need to be performed and at which time this should be done such that the service tasks of each train unit in a departing train are completed before the departure time, and none of the resource capacity constraints are exceeded in the shunting plan. A more in-depth description of the scheduling problem can be found in [1, 2].

To handle common disruptions during the operational phase, such as a train that arrives late, or a service activity that takes longer than expected, the shunting plan is used as an initial baseline schedule, and a rescheduling policy is applied for on-line adjustments to the plan. The baseline schedule is represented as a *partial ordering schedule (POS)* of the activities, specifying precedence relations that ensure the *resource feasibility* of any plan execution. The rescheduling policy is an *earliest start time (EST)* policy, which assigns each activity to its earliest possible starting time in the baseline schedule, and, in case of disruptions during operation, delays activities that have not

yet started as much as necessary while maintaining the ordering in the baseline schedule.

The Dutch Railways prefers robust shunting plans that require little rescheduling during the operational phase to avoid cascading effects. The robustness of a shunting plan can be measured as the probability that disturbances result in delayed train departures. Determining the probability of delays in schedules with precedence constraints is a computationally hard problem. To obtain a quick estimate of the robustness, a common approach is to use efficiently computable schedule characteristics, known as *robustness measures*, that show a strong correlation with the performance metric of interest. In [3], we have identified several robustness measures that accurately estimate the robustness of *randomly generated* shunting plans.

In this research, we study the impact of including robustness measures in the objective function of a local search algorithm on the robustness of the generated solutions. A large number of shunting plans are generated with different objective functions for real-world instances of a shunting yard operated by the NS, and the robustness of these plans is approximated using Monte Carlo simulation to identify which robustness measures guide the local search to highly robust solutions.

2 Robustness Measures

Many robustness measures proposed in literature, such as the measures found in [7], [4], [5], and [6], are based on the concept of *slack*, which quantifies the maximal allowed delay of an activity in the schedule. The *total slack* is the amount of time an activity in the partial order schedule can be delayed without exceeding any deadline in the schedule, whereas the *free slack* is the amount of delay that an activity can have before it starts delaying any of its successors in the schedule.

In [3], we present an alternative to the slack-based measures. Elaborating on the approach from [9], we approximate the distribution of the completion time of each event in topological order. We assume that the completion time of each event i follows a normal distribution, for which we estimate the expected value and variance. The start time of event i is equal to the maximum completion time of the events immediately preceding i; given the expected values and variances of the completion times of these predecessors, we estimate the expected value and variance of the start time of i by iteratively using the formulas derived by [8]. Finally, we add the processing time of i to find the completion time. We call this method the *normal estimation*.

3 Local Search

In [1, 2] we have developed a local search algorithm based on Simulated Annealing that can find feasible shunting plans for real-life problem instances with deterministic data. Starting with an infeasible initial solution, the local search algorithm iteratively modifies the current solution to resolve conflicts in the shunting plan. These modifications consist of small, local changes to the

schedule, such as permuting the order of train movements or service activities, changing the parking location of trains, moving a train to another parking location, and assigning service activities to other facilities. We use this algorithm as our basis.

The objective function described in [2] focuses primarily on the conflicts in the shunting plan; the goal is to minimize the penalties incurred due to violations of the hard constraints. A secondary objective is the minimization of the number of train movements in the solution. Although the insertion of an additional train movement is occasionally necessary to resolve a conflict in the shunting plan, adding many movements generally hinders the search for feasible solutions.

Since the focus of this objective is limited to the feasibility of the shunting plan, it will likely lead to solutions that handle disruptions poorly. One approach to steer the local search towards more robust solutions is to include a robustness measure in the objective function. In [3] we have identified three promising candidate measures that correlate strongly with the robustness of shunting plans, namely the *minimum total slack*, the *minimum free slack*, and the *normal estimation*. The weight of the robustness measure in the objective must be small enough to ensure that the local search will always prefer a feasible shunting plan over an infeasible, yet robust plan.

4 Preliminary Results

We have tested the performance of the local search with the different robustness objectives on a real-world instance of the shunting problem. The instance consists of 19 trains arriving at the *"Kleine Binckhorst"*, a shunting yard operated by NS, where the trains have to be cleaned, washed and maintained. A full description of the problem instance is provided in [3]. The main sources of uncertainty during shunting are the arrival time of trains and the duration of service and movement activities. In our experiments, we assume that trains arrive uniformly within a ten minute interval centered around their expected arrival time, and that duration of movement and service tasks follows a log-normal distribution with the nominal duration as the mean, and a standard deviation of 10% of the mean.

To study the effect of the robustness measures, we generated solutions using our local search by minimizing either only the conflict and movement penalties, or the base objective extended with one of the three robustness measures, resulting in 100 feasible shunting plans for each of the four configurations. The robustness of each of the plans — the probability of a delayed train departure — was estimated by sampling 20000 plan realizations in a Monte Carlo simulation.

Table 1 summarizes the results of the simulation. The columns **average**, **Min** and **Max** show statistics on the probability of delay in the solutions generated per objective function. The number of cases in which the local search failed to find a feasible solution within five minutes is listed in the **Failures** column, and the last two columns show the expected number of five minute runs of

Objective	**Average**	**Min**	**Max**	**Failures**	$p = 0.05$	$p = 0.01$
Basic	0.19	0	0.77	13	2.22	2.46
Total Slack	0.06	0	0.60	11	1.42	1.68
Free Slack	0.08	0	0.72	15	1.51	1.64
Normal Estimation	0.02	0	0.23	34	1.54	1.78

Table 1: The results of the Monte Carlo simulation of the solutions per objective function.

the local search needed until a feasible solution with a delay probability less than p is found.

The probability that the local search finds a feasible and robust solution for this shunting problem instance with the non-robust objective is surprisingly high, suggesting that a multi-start local search approach that restarts the search whenever no feasible, robust solution is found within five minutes, might be sufficient to generate robust shunting plans in reasonable time.

However, the results indicate that the probability of finding a robust solution can be improved significantly by adding any of the three robustness measures to the objective. Although the normal estimation method shows the lowest average and maximum delay probabilities, it requires considerably more time to evaluate a shunting plan in an iteration. The local search failed to find a feasible solution within five minutes, which is the maximum computation time that we allowed per solution, far more often with the normal estimation objective than with the other three configurations. Therefore, the minimum total slack is a viable alternative if the computation time is limited, as it performs only slightly worse than the normal estimation approach.

5 Conclusion

The aim of this research is to develop an efficient algorithm for finding robust solutions for the train shunting problem that arises at shunting yards, where the robustness is defined as the probability of delays in the scheduled train departures. We have compared several search objectives that incorporate robustness measures with a basic, non-robust objective in a Monte Carlo simulation of solutions generated with an existing local search for a real-world shunting problem of the NS. We have shown that the addition of a robustness measure based on estimating the completion times significantly improves the robustness of the solutions generated by the local search, outperforming minimum slack robustness measures. This improvement comes at the cost of an increase in computation time, and for the instance under consideration, it turns out that we can better spend this additional time on running the local search more often. A topic for further research is to investigate whether these results generalize to other problem instances, for which it is harder to find a feasible solution, or to a broader class of planning problems.

References

[1] R.W. van den Broek, J.A. Hoogeveen, and J.M. van den Akker, "Train Shunting and Service Scheduling: an integrated local search approach", Master thesis, Utrecht University, `https://dspace.library.uu.nl/handle/1874/338269` (2016).

[2] R.W. van den Broek, J.A. Hoogeveen, J.M. van den Akker, and B. Huisman, "A Local Search Algorithm for Train Unit Shunting with Service Scheduling", *Manuscript submitted for publication* (2018).

[3] R.W. van den Broek, J.A. Hoogeveen, and J.M. van den Akker, "How to Measure the Robustness of Shunting Plans", *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)* (2018).

[4] H. Chtourou and M. Haouari, "A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling", *Computers & industrial engineering* 55, 183-194 (2008).

[5] Ö. Hazir, M. Haouari, and E. Erdal, "Robust scheduling and robustness measures for the discrete time/cost trade-off problem", *European Journal of Operational Research* 207, 633-643 (2010).

[6] M.A. Khemakhem and H. Chtourou, "Efficient robustness measures for the resource-constrained project scheduling problem", *International Journal of Industrial and Systems Engineering* 14, 245-267 (2013).

[7] J.V. Leon, D.S. Wu, and R.H. Storer, "Robustness measures and robust scheduling for job shops", *IIE transactions* 26, 32-43 (1994).

[8] S. Nadarajah and S. Kotz, "Exact distribution of the max/min of two Gaussian random variables", *IEEE Transactions on very large scale integration (VLSI) systems* 16, 210-212 (2008).

[9] G. Passage and J.M. van den Akker and J.A. Hoogeveen, "Combining local search and heuristics for solving robust parallel machine scheduling", Master thesis, Utrecht University, `https://dspace.library.uu.nl/bitstream/handle/1874/334269/thesis.pdf` (2016)

Incorporating competition in demand-based optimization models

Stefano Bortolomiol
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne
Email: stefano.bortolomiol@epfl.ch

Virginie Lurkin
Department of Industrial Engineering and Innovation Sciences
Eindhoven University of Technology

Michel Bierlaire
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne

1 Introduction

Oligopolistic competition occurs often in transportation, due to reasons such as external regulations, limited capacity of the infrastructure and difficulty in entering a well-established market. In transport oligopolies, operators take the supply-side decisions that optimize their own objective function. Such decisions are influenced both by the preferences of the customers, who want to purchase one of the services on the market, and by the decisions of the competitors.

In this work, the preferences of the customers are modelled at a disaggregate level according to random utility theory and are embedded in each operator's optimization problem. Using a disaggregate approach that accounts for heterogeneous demand allows to better model supply-demand interactions. Competition among market players is modelled explicitly as a non-cooperative game in which all operators take into account the decisions of their competitors. This results in a multi-leader-follower game, for which a MIP model inspired by the fixed-point iteration algorithm is proposed to find Nash equilibrium solutions.

2 Demand-based optimization

By incorporating customer behavior inside their optimization problem, suppliers can improve many of their strategic decisions. Generally, demand-based optimization problems can be modelled as Stackelberg games [1]. Equivalent Stackelberg problems are frequent in transportation when a supplier or regulator knows the utility functions of its potential customers, who collectively play the follower role. From a modelling perspective, the result is an optimization problem having optimization problems in the constraints, also known as bilevel program [2, 3].

Applications of demand-based optimization models include revenue management [4, 5] and road tolling [6], among others. The majority of the papers propose nonlinear formulations and estimate choice probabilities with the multinomial logit model (MNL). To overcome MNL limitations on random taste variation or correlation between alternatives, more complex discrete choice models such as the nested logit and the mixed multinomial logit have also been used.

A framework that can integrate any discrete choice model in a MILP is presented in [7]. More specifically, choice probabilities can be linearized by using simulation to draw from the utility function's known error term distribution. For all customers and alternatives, a number of draws are extracted, corresponding to different behavioral scenarios. In each scenario customers deterministically choose the utility-maximizing alternative. Over multiple scenarios, the choice probability of an alternative is equal to the number of times the alternative is chosen over the number of draws.

Nonlinear and linear demand-based optimization models

Consider a set N of customers, a set I of alternatives and a set $I_k \subseteq I$ of alternatives managed by the supplier. Let V_{in} and P_{in} be the utility associated by customer $n \in N$ to alternative $i \in I$ and the corresponding choice probability. For the sake of simplicity, we assume that prices p_{in} are the only upper-level decision variables, that the supplier has no operational costs and that choice probabilities are estimated by using a MNL. The optimization problem is then

$$\max \quad \sum_{i \in I_k} \sum_{n \in N} p_{in} P_{in} \tag{1}$$

$$s.t. \quad P_{in} = \frac{\exp(V_{in})}{\sum_{j \in I} \exp(V_{jn})} \qquad \forall i \in I, \forall n \in N \tag{2}$$

$$V_{in} = \beta_{in} p_{in} + q_{in} \qquad \forall i \in I, \forall n \in N. \tag{3}$$

The objective function (1) maximizes the supplier's revenue. Constraints (2) derive the choice probabilities. Constraints (3) define the deterministic utility functions, composed of an exogenous term q_{in} and an endogenous term that depends on the price, which is the variable linking the upper-level problem with the lower-level problem.

For the linearized version of the model, let R be the set of behavioral scenarios. For each $r \in R$, an error term parameter ξ_{inr} is drawn from the known distribution. The variables $U_{nr} = \max_i U_{inr}$

capture the value of the highest utility for customer n in scenario r, while the binary decision variables w_{inr} identify the alternative i chosen by each customer n in each scenario r. Constraints (2-3) can be now written as

$$s.t. \quad P_{in} = \frac{\sum_{r \in R} w_{inr}}{R} \qquad \forall i \in I, \forall n \in N \tag{4}$$

$$U_{inr} = \beta_{in} p_{in} + q_{in} + \xi_{inr} \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{5}$$

$$U_{inr} \leq U_{nr} \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{6}$$

$$U_{nr} \leq U_{inr} + M_{U_{nr}}(1 - w_{inr}) \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{7}$$

$$\sum_{i \in I} w_{inr} = 1 \qquad \forall n \in N, \forall r \in R. \tag{8}$$

The utility functions (5) now includes a drawn error term. Constraints (6-7) ensure that in each behavioral scenario customers deterministically choose the alternative yielding the highest utility.

3 A MIP for the fixed-point problem

In oligopolistic markets there are multiple players that simultaneously solve a demand-based optimization problem. The result is a multi-leader-follower game in which the payoffs are a function of both the decisions of the customers and the strategies of the competitors. Given the complexity of the demand-based optimization framework, well-known results on the existence or uniqueness of pure or mixed strategy Nash equilibria cannot be exploited and alternative approaches are needed.

Several works dealing with Nash equilibria in transportation adopt an algorithmic approach based on the fixed-point iteration method (see for example [8] and [9]). Starting from an initial feasible solution to the problem, operators take turns to play their best response pure strategy to the last strategy played by the competitors. Such sequential game terminates when a solution is repeated, as it induces the same sequence of best responses as before. Such solution is either a Nash equilibrium for the game or a set of n strategies for each player, with $n > 1$, which would continue to be played cyclically.

Solving the multi-leader-follower game as a sequential game is attractive from a computational perspective. The sequential game is also easily interpretable, since it reproduces the behavior of two or more players that do not know the competitors' objective function. However, the convergence proof of such algorithm depends on conditions such as having a convex payoff function [10], which are not verified in the multi-leader-follower games we want to solve. Consequently, by solving the problem as a sequential game there is no guarantee of existence or uniqueness of a pure strategy Nash equilibrium. Finally, different initial solutions could lead to different equilibria.

We propose a new mathematical model to find equilibria in multi-leader-follower games. It models the sequential game as a one-step approach by considering only two iterations of the fixed-point problem. We define as *distance* between two solutions a non-negative value measuring the

difference in operators' decisions, in customers' decisions, or a combination. If we start from an equilibrium point, the distance between the initial solution and the next iteration solution is equal to 0. Else, the distance is greater than 0, since at least one of the players changes its strategy.

The notation is now introduced for the linear model. Let K be the set of the operators and let S_k be the given finite set of strategies that can be played by operator $k \in K$. The parameters p_{ins} indicate the price at which alternative i is offered to customer n by operator k if playing strategy $s \in S_k$. The superscripts $'$ and $''$ refer to the variables of the initial configuration and of the best response configuration, respectively. The variables V_s store the value of the payoff for operator k if responding with strategy $s \in S_k$, while the variables V_k^{max} store the highest of these values for each operator. The binary variables x_s are equal to 1 if strategy $s \in S_k$ is the best response of operator k to the initial configuration. Then, the mathematical model can be written as

$$\min \sum_{i \in I} \sum_{n \in N} |p_{in}^{''} - p_{in}^{'}| \tag{9}$$

$s.t.:$

Initial configuration:

$$\sum_{i \in I} w_{inr}^{'} = 1 \qquad \forall n \in N, \forall r \in R \tag{10}$$

$$U_{inr}^{'} = \beta_{in} p_{in}^{'} + q_{in} + \xi_{inr} \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{11}$$

$$U_{inr}^{'} \leq U_{nr}^{'} \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{12}$$

$$U_{nr}^{'} \leq z_{inr}^{'} + M(1 - w_{inr}^{'}) \qquad \forall i \in I, \forall n \in N, \forall r \in R \tag{13}$$

Final configuration:

$$\sum_{i \in I} w_{inrs}^{''} = 1 \qquad \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{14}$$

$$U_{inrs}^{''} = \beta_{in} p_{ins} + q_{in} + \xi_{inr} \qquad \forall i \in I_k, \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{15}$$

$$U_{inrs}^{''} = U_{inr}^{'} \qquad \forall i \in I \setminus I_k, \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{16}$$

$$U_{inrs}^{''} \leq U_{nrs}^{''} \qquad \forall i \in I, \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{17}$$

$$U_{nrs}^{''} \leq z_{inrs}^{''} + M(1 - w_{inrs}^{''}) \qquad \forall i \in I, \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{18}$$

Best response constraints:

$$p_{in}^{''} = \sum_{s \in S_k} p_{ins} x_s \qquad \forall i \in I_k, \forall n \in N, \forall k \in K \tag{19}$$

$$V_s = \frac{1}{R} \sum_{i \in I_k} \sum_{n \in N} \sum_{r \in R} p_{ins} w_{inrs}^{''} \qquad \forall s \in S_k, \forall k \in K \tag{20}$$

$$V_s \leq V_k^{max} \qquad \forall s \in S_k, \forall k \in K \tag{21}$$

$$V_k^{max} \leq V_s + M(1 - x_s) \qquad \forall s \in S_k, \forall k \in K \tag{22}$$

$$\sum_{s \in S_k} x_s = 1 \qquad \forall k \in K \tag{23}$$

$$w_{inrs}^{'}, w_{inrs}^{''} \in \{0, 1\} \qquad \forall i \in I, \forall n \in N, \forall r \in R, \forall s \in S_k, \forall k \in K \tag{24}$$

$$x_s \in \{0, 1\} \qquad \forall s \in S_k, \forall k \in K. \tag{25}$$

The objective function (9) minimizes the distance between the two solutions in terms of operators'

strategies. The absolute value can be linearized by expressing the argument as the difference of two non-negative variables and by minimizing the sum of these variables in the objective function. Constraints (10-13) define the utilities and force customers to always choose the alternative with the highest utility in the initial configuration. Notice that the price variables p'_{in} are modeled as free continuous variables. Constraints (14-18) impose the utility maximization principle in the best response configurations. In each strategic scenario, the decisions of the optimizing operator only affect the utility of its alternatives (15), while the utility of the competitors' alternatives remain unchanged (16). Finally, constraints (19-23) state that operators always select the best response strategy to the initial configuration.

Compared to the sequential game, this model enables discrimination between different equilibrium solutions by modifying the objective function, and it can also find near-equilibrium solutions, if no Nash equilibrium exists. It can be also applied to a nonlinear model having probabilistic customer choices. The description of the nonlinear case is omitted here.

4 Numerical experiments

The case study used for the tests is derived from [11], where the choice of customers among three different parking alternatives is modelled with a mixed logit model. The nonlinear and the linear optimization models for both the Stackelberg game and the multi-leader-follower game were tested on two discrete choice specifications, namely the multinomial logit model and the mixed logit model. Extended results of the experiments are available online [12].

The experiments show that for the Stackelberg game the nonlinear model converges to optimality much faster than the MILP model in all cases. For the multi-leader-follower game the nonlinear model converges faster to optimality only for the logit formulation, as there is no need for simulation. On the other hand, when using a mixed logit formulation, the linear model generally outperforms the nonlinear model, which fails to converge on larger instances. The worsening of the computational performance of the nonlinear model in the competitive case can be imputed to the discretized price parameters and to the binary decision variables of the upper-level problems, while the improved performance of the MILP model can be explained by the reduction of the solution space due to the limited set of response strategies. In particular, the linear model, which is structured around a simulation framework, has similar computational performances on the logit and the mixed logit model. The latter finding is particularly encouraging, because it indicates that the MILP formulation for the demand-based optimization model could potentially embed even the most complex and accurate discrete choice models.

5 Future research

The numerical experiments performed so far indicate that different formulations could be more or less effective depending on the type of decision variables and on the chosen discrete choice model. The nonlinear formulation is non-convex and becomes intractable when many discrete variables are introduced, while the linear formulation is convex but combinatorial due to the nature of the simulation framework.

In the next phases of this research, we plan to (i) write the Stackelberg game as a mathematical program with equilibrium constraints (MPEC) and the multi-leader-follower game as an equilibrium program with equilibrium constraints (EPEC), to investigate whether continuous formulations can be helpful to find solutions to our initial problem or to improve the bounds of the fixed-point MIP model, and (ii) propose an algorithmic framework in which candidate equilibrium solutions are found by means of different heuristic blocks and used as input strategies in the fixed-point MIP model.

References

[1] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. Julius Springer, 1934.

[2] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

[3] Stephan Dempe and Joydeep Dutta. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Mathematical programming*, 131(1-2):37–48, 2012.

[4] S-E Andersson. Passenger choice analysis for seat capacity control: A pilot project in Scandinavian Airlines. *International Transactions in Operational Research*, 5(6):471–486, 1998.

[5] Kalyan Talluri and Garrett Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

[6] Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management science*, 44(12-part-1):1608–1622, 1998.

[7] Meritxell Pacheco Paneque, Shadi Sharif Azadeh, Michel Bierlaire, and Bernard Gendron. Integrating advanced discrete choice models in mixed integer linear optimization. Technical report, Transport and Mobility Laboratory, EPFL, 2017.

[8] Nicole Adler. Competition in a deregulated air transportation market. *European Journal of Operational Research*, 129(2):337–345, 2001.

[9] Andrew Koh and Simon Shepherd. Tolling, collusion and equilibrium problems with equilibrium constraints. *European Transport*, 44:3–22, 2010.

[10] Jong-Shi Pang and Donald Chan. Iterative methods for variational and complementarity problems. *Mathematical programming*, 24(1):284–313, 1982.

[11] A Ibeas, L Dell'Olio, M Bordagaray, and J de D Ortúzar. Modelling parking choices considering user heterogeneity. *Transportation Research Part A: Policy and Practice*, 70:41–49, 2014.

[12] Stefano Bortolomiol. Incorporating competition in demand-based optimization models. Numerical experiments. `https://enacshare.epfl.ch/fEBjNDym7YJ8ATvxwrtPgRq2af5CoHV`, 2018. Online, last accessed on 12 October 2018.

Passenger-centric dial-a-ride problem for on-demand mobility systems

Shadi Sharif Azadeh

Department of Econometrics, Erasmus University Rotterdam

Yousef Maknoon

Department of Technology and Policy Management, Delft University of Technology

Bilge Atasoy

Department of Maritime and Transport Technology, Delft University of Technology

Michel Bierlaire

Department of Civil and Environmental Engineering, EPFL

Moshe Ben Akiva

Department of Civil and Environmental Engineering, MIT

Email: sharifazadeh@ese.eur.nl

1 Introduction

Constant increase in traffic congestion for urban areas and the cost of expanding infrastructure as well as lack of space (e.g., in Europe) in the last couple of decades, have increased congestion and pollution in cities. Having a better public transit is not a sufficient solution for suburban areas where the ridership is low, and profitability decreases due to under-utilization of the transportation capacity. As a result, transport planners seek to find solutions to use the current infrastructure more efficiently.

This paper presents an algorithmic framework to solve passenger-centric (using discrete choice models) dial-a-ride assortment optimization problem. This problem belongs to the class of demand responsive systems in which a customized travel service is offered to each passenger [1]. When a passenger arrives at the system, he/she requests a ride service. Each request contains information about the pickup/delivery locations and preferred pickup time. The system then provides a set of ride options (i.e. travel menu) to the passenger. Ride options are created based on the proposed pickup time, type of service (e.g. taxi/shared-taxi), charging the fare and maximum duration of

the trip (in case of shared-taxi). We measure the attractiveness of each ride option with a utility function and use multinomial logit to model passengers behavior. The optimization problem is then defined by constructing a travel menu that maximizes the expected profit. As the cost of ride options are not known a priori, the expected maximum profit is achieved by simultaneously solving the assortment optimization (to determine the sub-set of ride options) with the dial-a-ride problem (to calculate the underlying routing cost).

This problem is defined as a unique combination of two well-known problems: (1) dial-a-ride problem and (2) assortment optimization. All studies in dial-a-ride problem focus on finding a minimum-cost route that satisfies all customers demand (see [2]; [3]). In this problem, we relax this constraint. Here, the selection of ride options is determined according to the assortment optimization. Unlike, all studies in choice-based assortment optimization (see [4]) the cost of each option is not known a priori and has to be calculated by solving the inherited dial-a-ride problem.

2 Problem description

A company is responsible for transferring a set of n customer requests ($r \in R, |R| = n$) with a set of identical vehicles. Requests $1, \ldots, n-1$ are existing ones and n is a request of the new customer. For existing customers, the pickup and delivery location, pickup time-window and maximum ride time are known. For the new customer, information about the pickup/delivery locations and preferred pickup time has entered the system. Based on the new request, a set of ride options are generated. These options are generated based on the variation of pickup time, types of the service (e.g. taxi, shared-taxi) and fare level. We present the routing problem as a complete directed graph. For each request, two nodes are defined to present its pickup and delivery locations. Similarly, with each arc, two non-negative values are associated: a travel cost and a seat opportunity cost. The later is used to quantify the economic value of a seat for the traveling period.

We formulate the joint routing-assortment optimization problem as a mixed-integer linear model. The objective of the model is to maximize the expected revenue with two blocks of the constraints. The first block of the is related to the assortment problem. These constraints are linked to those in the second block where explicit routing decision has to be made. Here, we use the three-index binary variable to model vehicle route.

3 Solution approach

To solve the mathematical model, we have developed a branch& cut algorithm. We first investigate the structural properties of the assortment problem. Based on that, a series of valid inequalities are generated and added to the model. Other valid inequalities used to solve dial-a-ride problem

are also included (see [5]). A pre-processing approach (based on the structural properties of the problem) is developed to reduce the number of variables and constraints.

For a dynamic problem, we solve the mathematical model every time a new user arrives. The mathematical model determines the optimal assortment offered to the customer. Thereafter, we simulate the customer choice and update the transportation plan. This process continues till the end of the planning horizon.

4 Results

The algorithm was implemented in C++ and we use CPLEX 12.6 as our mathematical solver. We evaluate the performance of the algorithm on a set of instances with 25 and 50 requests. The computational results are performed in two parts.

In the first part, we choose instances with 25 requests and focus on demonstrating the performance of the branch & cut algorithm. We consider the off-line problem (all requests are known in advance). For each instance, we evaluate the impact of each B&C components (e.g. valid inequality, pre-processing, etc) on the performance (computation time, number of branched nodes and integrality gap) of the algorithm. Moreover, we compare our B&C method with the one that only use CPLEX as a general solver. The results denote significant reduction in computational time (around 38% on average).

In the second part, we focus on dynamic problems to evaluate its revenue performance. We choose instances with 50 requests that dynamically arrive at the system. All instances have the same pickup and delivery coordinates and are varied based on the following attributes: (1) arrival frequency (frequent, infrequent), (2) advanced booking (up to 20 minutes in advance, between 20 to 60 minutes in advance) and (3) opportunity cost (high and low). The results are summarized in Table 1 (two parts). For each part, the number of served users (out of 50), the average profit per user, and the average number of users on-board are reported.

Table 1 (part 1), presents the computational results of the instances with loose booking (between 20 to 60 minutes in advance). As can be seen, the on-demand system serves fewer users in total for the case of frequent arrival. However, the average profit is slightly more regardless of the opportunity cost. For the case of frequent arrivals, we further denote the trade-off between frequency and opportunity cost. As can be seen in Part 2, the number of served users, as well as profit per user, increased as the system has more flexibility to operate. That means the advance booking enables the system to have a better fleet management which leads to lower operating cost and improved profit. By introducing the opportunity cost for empty seats, we penalize empty cars circulations in the network which results in reducing the negative environmental impacts and congested routes.

Table 1: Experimental Results

Part 1: Frequent vs infrequent arrival - high vs low opportunity cost (loose booking)

	Infreq arr. High opp. cost	Freq arr. High opp. cost	Infreq arr. Low opp. cost	Freq arr. Low opp. cost
Served users	49.20	42.50	47.4	34.8
Profit per user	9.95	11.36	10.32	10.74
Avg users on-board	1.20	1.60	1.00	1.00

Part 2: Loose vs tight booking - high vs low opportunity cost (frequent arrival)

	Loose booking High opp. cost	Tight booking High opp. cost	Loose booking Low opp. cost	Tight booking Low opp. cost
Served users	42.50	39.00	34.80	32.80
Profit per user	11.36	10.70	10.74	10.71
Avg users on-board	1.60	1.39	1.00	1.00

References

[1] Atasoy B, Ikeda T, Song X, Ben-Akiva ME "The concept and impact analysis of a flexible mobility on demand system." Transportation Research Part C: Emerging Technologies, 56:373–392 (2015).

[2] Ho, Sin C., et al. "A survey of dial-a-ride problems: Literature review and recent developments." Transportation Research Part B: Methodological (2018).

[3] Cordeau, Jean-Franois, and Gilbert Laporte. "The dial-a-ride problem: models and algorithms." Annals of operations research 153.1 (2007): 29-46.

[4] Kk, A. Grhan, Marshall L. Fisher, and Ramnath Vaidyanathan. "Assortment planning: Review of literature and industry practice." Retail supply chain management. Springer, Boston, MA, 2008. 99-153.

[5] Cordeau, Jean-Franois. "A branch-and-cut algorithm for the dial-a-ride problem." Operations Research 54.3 (2006): 573-586.

Railway Rolling Stock Maintenance Scheduling

Lukas Bach, Daniel Palhazi Cuervo
Optimization, SINTEF Digital, Oslo, Norway
Email: lukas.bach@sintef.no

1 Motivation

The railway sector is one of the foundations of a country's mobility. Only in Europe, rail represents 45% of public transport, accounts for 26.9 billion passenger trips each year and is used to carry 18.3% of the freight goods inland [2]. One of the most prominent costs drivers in this sector is the maintenance of the rolling stock. Maintenance costs are estimated to average between 2.5 and 3.5 euros per rolling-stock unit, per kilometer travelled [3]; accounting for more than 30% of the total costs of operating a train.

In this presentation, we discuss an ongoing research project in collaboration with Mantena, the main provider of rolling stock maintenance in Norway. This company has more than 1100 employees and hosts operations in around 13 locations. In order to keep a competitive position, Mantena requires a very detailed planning of its operations. This involves, among other things, scheduling each maintenance task (that needs to be carried out on each rolling-stock unit) and determining the number of employees required to perform them. Such a planning/scheduling problem not only should minimize the overall cost, but take into account multiple operational constraints (i.e., capacity and equipment availability), ensure that trains comply with safety regulations, and guarantee that employment contracts adhere to labor directives and collective agreements. Previous attempts to solve rolling stock maintenance problems are often focused on the rolling stock rostering and routing to receive maintenance, see [1] for a recent approach and overview. In this work we focus on when to maintain a specific train and which maintenance tasks to perform.

2 Problem description

Each rolling stock has a *maintenance program* associated to it: a collection of (maintenance) *activities* that need to be carried out for the rolling stock to remain operative. Each activity, in turn, defines so-called *execution cycles*: sets of similar tasks to be executed periodically on a specific location of the rolling stock. Performing (the tasks within) a cycle requires an estimated

number of man hours, along with materials and other spare parts. For instance, passenger trains usually involve an activity to verify the proper functioning of the boogies that support each wagon. This activity involves, at least, three different cycles: 1) a light examination, that requires just a few man-hours, to be carried out every 250 000 km, 2) a more ellaborate check-up to be carried out every 1 250 000 km and 3) an thorough inspection, that demands several man-days of work, to be carried out every 3 750 000 km. Logically, the execution of cycle 3 is assumed to cover the tasks within cycles 2 and 3. Therefore, when a wagon is close to reach the mark of 3 750 000 km, only cycle 3 should be scheduled to be carried out.

In general, the optimization problem faced by Mantena involves scheduling the execution of each cycle (of each maintenance activity) in such a way that the overall operating cost is minimized. In a simplified version of the problem (outlined for illustrative purposes), this is achieved by scheduling each cycle as close to its deadline as possible. The objective is therefore to minimize the overall "earliness" of the schedule. A time-index formulation of this problem, based on that proposed by [4], is sketched below. The set of scheduling constraints are defined by Equations (2)-(7), connecting constraints are defined by Equations (8)-(11), capacity constraints by Equation (12) and integrality constraints by Equations (13)-(14).

Constant values

L	Set of types of labor available
A	Set of maintenance activities
C_a	Set of cycles of maintenance activity a
$\hat{C}_{ac}$	Set of cycles (of activity a) the executions of which cover cycle c
m_{acl}	Number of man hours of type of labor l, required to execute cycle c of activity a
P	Set of time periods
f_{ac}	Frequency of execution of cycle c of activity a (maximum number of time periods between two consecutive executions)
e_{ac}	The initial execution period of cycle c of activity a
a_l^p	Number of man hours of type of labor l available at time period p

Decision variables

x_{ac}^p	1 if cycle c of activity a is executed in time period p, 0 otherwise
y_{ac}^p	1 if the due period of cycle c of activity a is covered in time period p, 0 otherwise
z_{ac}^{pq}	1 if p and $p+q$ are consecutive execution (or covered) periods of cycle c of activity a, 0 otherwise.

Objective function

$$\min \sum_{a \in A} \sum_{c \in C_a} \sum_{p=e_{ac}}^{|P|-f_{ac}} \sum_{q=1}^{f_{ac}} (f_{ac} - q) z_{ac}^{pq} \tag{1}$$

In constraint (2) we ensure that a cycle of a maintenance task cannot be scheduled to be executed before its initial execution period. The first execution of a cycle (of an activity) must be scheduled in its initial execution period, this is guaranteed by constraint (3). Constraint (4), at most, one cycle of an activity can be scheduled to be executed in a time period. In constraint

Constraints

$$x_{ac}^{p} = 0, \quad \forall a \in A, c \in C_a, p \in \{1, \ldots, e_{ac} - 1\} \tag{2}$$

$$x_{ac}^{e_{ac}} = 1, \quad \forall a \in A, c \in C_a \tag{3}$$

$$\sum_{c \in C_a} x_{ac}^{p} \leq 1, \quad \forall a \in A, p \in P \tag{4}$$

$$\sum_{q=1}^{f_{ac}} z_{ac}^{e_{ac},q} \geq 1, \quad \forall a \in A, c \in C_a \tag{5}$$

$$x_{ac}^{p} + y_{ac}^{p} - \sum_{q=1}^{f_{ac}} (x_{ac}^{p+q} + y_{ac}^{p+q}) \leq 0, \quad \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P| - f_{ac}\} \tag{6}$$

$$\sum_{p=|P|-f_{ac}}^{|P|} x_{ac}^{p} \leq 1, \quad \forall a \in A, c \in C_a \tag{7}$$

$$\sum_{\hat{a} \in \hat{C}_{ac}} x_{\hat{a}c}^{p} - y_{ac}^{p} = 0, \quad \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P|\} \tag{8}$$

$$x_{ac}^{p} + y_{ac}^{p} \leq 1, \quad \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P|\} \tag{9}$$

$$x_{ac}^{p} + y_{ac}^{p} - \sum_{q=1}^{f_{ac}} z_{ac}^{pq} = 0, \quad \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P| - f_{ac}\} \tag{10}$$

$$\begin{aligned} x_{ac}^{p} + y_{ac}^{p} + x_{ac}^{p+q} + y_{ac}^{p+q} \leq \sum_{r=1}^{q-1} (x_{ac}^{p+r} + y_{ac}^{p+r}) + z_{ac}^{pq} + 1, \\ \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P| - f_{ac}\}, q \in \{1, \ldots, f_{ac}\} \end{aligned} \tag{11}$$

$$\sum_{a \in A} \sum_{c \in C_a} x_{ac}^{p} m_{acl} - a_{l}^{p} \leq 0, \quad \forall p \in P, l \in L \tag{12}$$

$$x_{ac}^{p}, y_{ac}^{p} \in \{0, 1\}, \quad \forall a \in A, c \in C_a, p \in P \tag{13}$$

$$z_{ac}^{pq} \in \{0, 1\}, \quad \forall a \in A, c \in C_a, p \in \{e_{ac}, \ldots, |P| - f_{ac}\}, q \in \{1, \ldots, f_{ac}\} \tag{14}$$

(5), the first scheduled execution, or due period reset, of a cycle (of an activity) must take place after its initial execution period and before its due period. The number of periods between two consecutive executions or resets of each cycle, of each activity, must never exceed its due period. This is addressed by (6). Constraint (7), there can be, at most, one scheduled execution of a cycle (of an activity) in the last part of the time horizon where the objective function can be calculated. Constraint (8), the due period of a cycle, of an activity, is reset if and only if any of its dominating cycles (those that reset its due period) is executed. A cycle, of an activity, cannot be executed and reset at the same period, this is enforced by constraint (9). Constraint (10) guarantees that if the cycle of an activity is scheduled to be executed or its due period reset, there must be only one consecutive execution or reset period before its due date. Two periods are consecutive execution or reset periods of a cycle (of an activity), if an only if the cycle has been scheduled to be executed or reset on those periods, and there is no scheduled execution or reset period in between. This is enforced by constraint (11) In constraint (12), the number of man hours (of a certain labor

type) required for a time period does not exceed the number of man hours available at the depot. Integrality is enforced by constraints (13) and (14).

3 Preliminary experiments

We will present an extension of the previous model that takes into account a more realistic objective function and a wider range of operating constraints. This extension considers, in addition, the trade-off between pre-poning the execution of cycles and hiring additional employees. We will also describe preliminary computational experiments that have been carried out to support Mantena's long-term planning. The results obtained have provided valuable input for the company to reduce the number of employees required to carry out its operations, and smoothen the workload on its maintenace depots.

References

[1] G. L. Giacco, D. Carillo, A. DAriano, D. Pacciarelli and . G. Marn. "Short-term Rail Rolling Stock Rostering and Maintenance Scheduling". *Transportation Research Procedia*, 3, 651-659, 2014.

[2] ERRAC, the European Rail Research Advisory Council. *Rail 2050 Vision - Rail, the Backbone of Europe's Mobility*, 2017.

[3] D. Gattuso, A. Restuccia. "A tool for railway transport cost evaluation". *Procedia-Social and Behavioral Sciences*, 111, 549-558, 2014.

[4] Núñez-del-Toro, C., Fernández, E., Kalcsics, J., Nickel, S. "Scheduling policies for multi-period services". *European Journal of Operational Research*, 251(3), 751-770, 2016.

Specification of Mixed Logit Models Assisted by an Optimization Approach

Alexander Paz
Transportation Research Center
University of Nevada, Las Vegas, U.S.
Email: apaz@unlv.edu

Cristian Arteaga
Transportation Research Center
University of Nevada, Las Vegas, U.S.
Email: arteagas@unlv.nevada.edu

1 Introduction

The modeling and prediction of discrete outcomes is a common problem in many areas, including economics, engineering, and medicine. Some examples of discrete outcome problems include (i) analysis of transportation modes (i.e., car, transit, or walking) based on observed socioeconomic characteristics; (ii) estimation of the presence of a pathology based on attributes of a patient; and (iii) estimation of how many cars will be owned based on observed characteristics of a household.

Mixed logit models have been proposed [1] as one of the most prominent techniques for modeling discrete outcome problems. Mixed logit models address the limitations of previous techniques by allowing modeling of variables with random coefficients. Such variables can follow any statistical distribution specified by the researcher as well as a general random term that follows an extreme value distribution. The predictive power and quality of a mixed logit depends greatly on an appropriate definition of the distribution of the random coefficients [2]. Given a mixed logit estimation problem, several assumptions are required to determine the best model specification. In general, the distribution of the random coefficients and potential explanatory variables need to be assumed before a model is estimated. This model specification process relies greatly on human judgment to include context-specific knowledge in the model and to accommodate interpretation needs. This process is time consuming and subject to expert knowledge and ad hoc trial-and-error approaches. Therefore, approaches that support the search for model specifications can help the analyst and decrease the time and effort required for this process

Harmony search is a metaheuristic that imitates the music improvisation process. This technique has been successfully applied to optimization problems in recent years [3]. When musicians compose a harmony, they usually combine various possible music notes stored in their memory. This search for a perfect harmony is comparable to an optimization process. One of the main advantages of this technique is that the hyper parameter selection is relatively easy and even if some of them are not perfectly set, the algorithm is still able to find good quality solutions [4]. This technique has also been applied to variable selection approaches [5].

This study proposes an approach to assist researchers with the specification of mixed logit models by optimizing the goodness of fit. The specification includes the variables considered as well as the distribution and associated parameters for the corresponding coefficients. A solution algorithm was implemented and tested with one dataset.

2 Methodology

The following notation is used to describe and formulate the proposed problem:

X	vector of potential explanatory variables
N	number of observations
K	number of potential explanatory variables
S	number of included variables
J	number of alternatives or discrete outcomes
i	subscript to denote a decision maker; i = 1, 2, …, N
j	superscript to denote an alternative; j = 1, 2, …, J
k	subscript for a variable, k = 1, 2, ..., K
y_{ij}	indicator variable equal to 1 if decision maker i chooses alternative j; 0 otherwise.
s_k	indicator variable to denote if x_k is included, $s_k \in \boldsymbol{s}$. s_k is 1 if variable x_k is included; 0 otherwise.
β_k^j	coefficient for variable x_k and alternative j; $\beta_k^j \in \boldsymbol{\beta}$.
$\boldsymbol{s}$	vector of included variables.
$\boldsymbol{\beta}$	vector of coefficients for potential explanatory variables.
$\boldsymbol{f}$	vector of density functions for coefficients $\boldsymbol{\beta}$.
f_k	density function for coefficient β_k. Possible density functions f_k are: normal (n), uniform (u), triangular (t)

The observed utility V_{ij} that a decision maker i obtains from alternative j can be represented as a linear dependency on the attributes of the decision maker and the alternatives as:

$$V_{ij} = \beta_0^j + \beta_1^j x_{i1} s_1 + \cdots + \beta_K^j x_{iK} s_K \tag{1}$$

For this research, the observed portion of utility V_{ij} was extended to add the indicator s_k of included variables. In mixed logit, log likelihood for the choices in a dataset is modeled as:

$$LL = \sum_{i=1}^{N} \sum_{j=1}^{J} y_{ij} \ln\left(\int \frac{e^{V_{il}}}{\sum_{j=1}^{J} e^{V_{ij}}} \boldsymbol{f}(\boldsymbol{\beta}) d\boldsymbol{\beta}\right) \tag{2}$$

The coefficients $\boldsymbol{\beta}$ can be estimated by maximum log- likelihood estimation (MLE). The objective is to find the model specification $\boldsymbol{M} = \{\boldsymbol{s}, \boldsymbol{f}\}$ with included variables $\boldsymbol{s}$ and the density functions $\boldsymbol{f}$ that minimize the Bayesian Information Criteria (BIC); this was expressed as:

$$\text{Min BIC} = \ln(N)\, S - 2\ln(LL) \tag{3}$$

A Harmony Search algorithm was used to solve the above minimization problem. Figure 1 illustrates the implemented steps. In the first step, the parameters of the algorithm and the harmony memory (HM) are initialized. The HM is sorted to keep track of the exact position of the best and worst solutions. In the second step, a new harmony H is improvised. For each variable, its value is assigned by taking it from the harmony memory or randomly depending on the Harmony Memory Consideration Rate (HMCR). Then, the pitch adjustment is performed for the variable *i* with Pitch Adjustment Rate (PAR) probability. For this study, the pitch adjustment performed by just taking a random value from the domain of the variable. In the third step, the new solution is added to the harmony memory if its BIC is smaller than the worst solution in the harmony memory. In the fourth step, the optimization process is stopped if there have been more than 150 iterations, otherwise, the step two starts again to generate another new harmony.

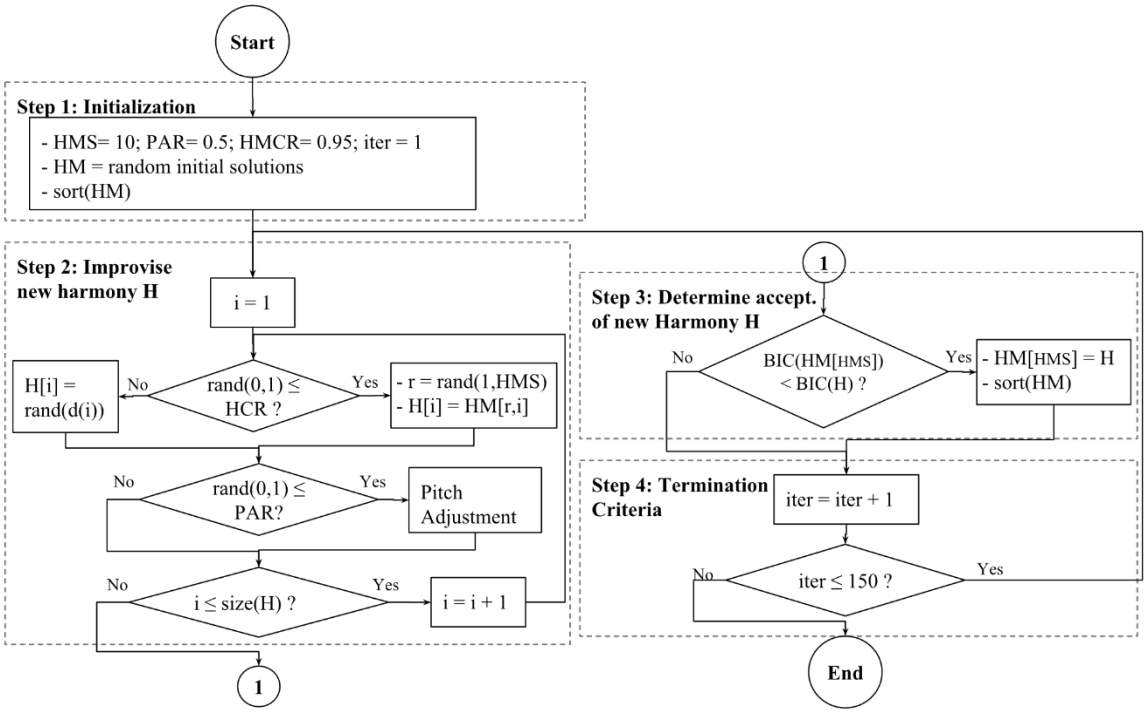


Figure 1. Implemented Harmony Search algorithm

3 Experiments and Results

A dataset with 4308 choices of electricity plans was used in this study. This dataset was initially used by Revelt & Train,1999 [6]. The mixed logit model specified by these authors is compared with the results of this study. Figure 2 illustrates the minimization in the BIC performed by the proposed approach. The BIC was lowered from 7971 to 7928.

Table 1 shows the model specification found by the proposed approach and the specification of Revelt & Train. It is noticeable that the coefficients between the two specifications are very similar in sign and magnitude, and the proposed technique was able to find a model with even lower BIC.

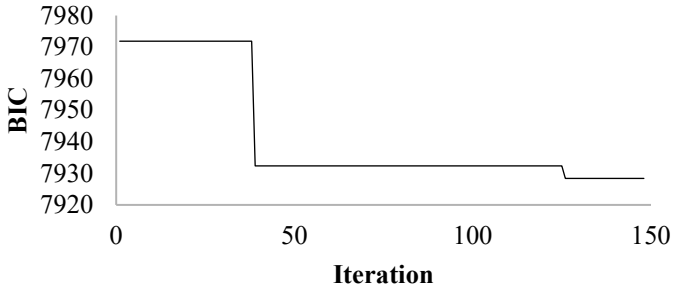


Figure 2. BIC vs Iterations for the implemented algorithm

Table 1. Model specification found by the proposed approach and specification by Revelt & Train

	Model by Revelt & Train			**Model found by proposed approach**			
	Coefficient	**Std. Error**	***f***	**Coefficient**	**Std. Error**	***f***	**Ratio Coeff.**
Price	-0.9080	0.0335		-0.9532	0.0345		0.95
Length of contract	-0.2503	0.0147		-0.2432	0.0143		1.03
Local supplier	2.2155	0.0839		2.1449	0.0816		1.03
Well known supplier	1.5687	0.0675		1.3239	0.0661		1.18
Time of day rates	-8.7502	0.2879		-9.0969	0.2914		0.96
Seasonal rates	-8.9501	0.2905		-8.9301	0.2896		1.00
Random Effects							
Price				0.2096	0.0107	n	
Length of contract	0.4156	0.0199	n	0.3806	0.0182	n	1.09
Local supplier	1.6423	0.0957	n	1.7384	0.0898	n	0.94
Well known supplier	1.0136	0.0750	n	1.7089	0.1160	t	0.59
Time of day rates	2.5623	0.1140	n	2.4488	0.1094	n	1.05
Seasonal rates	2.0030	0.1058	n	2.3803	0.1995	t	0.84
Log likelihood		**-3941.9**			**-3914**		
BIC		**7975.93**			**7928.393**		

4 Conclusions

The results suggest that the proposed algorithm can find an adequate specification for a mixed logit model in terms of goodness of fit, thereby assisting the analyst in the selection process. However, it is necessary to consider the judgement of the analyst to avoid suppression of variables or random effects that might be important for the interpretation of the model. This could be handled by adding constraints to guarantee the inclusion of elements defined by the analyst. The specifications found by the proposed approach are not necessarily final, rather they can be used to confirm or discard ideas or assumptions about the data generation process behind the problem. Such specifications can reveal hidden information or patterns that were not visible based on the problem context and available data. At the end of the optimization process, the harmony memory includes a set of specifications with low BIC values that can also be used by the analyst. That is, the analyst is not limited by one specification, but he/she has a range of specifications that can be combined with their expertise to produce a final useful model. In addition, the proposed approach can help reduce potential bias from the analyst

because the search strategy is based on finding the model that best fits the data and the objective function. This is very helpful in minimizing cases where models are forced to produce results that support a hypothesis.

References

[1] Train, K.E.. “Discrete Choice Methods with Simulation”. Cambridge University Press, Cambridge. (2009)

[2] Hensher, D.A., Greene, W.H., 2003. “The mixed logit model: The state of practice”. Transportation (Amst). 30, 133–176. (2003)

[3] Geem, Z. W., Kim, J. H., & Loganathan, G. V. “A new heuristic optimization algorithm: harmony search”. Simulation, 76(2), 60-68. (2001)

[4] Puri, T., & Ganguli, “Effect of User Defined Parameters of Harmony Search Algorithm (HSA) for Unconstrained Optimization Problems.” International Journal of Current Engineering and Scientific Research. Vol. 2. Issue 6. (2015)

[5] Sarvari, H., Khairdoost, N., & Fetanat, A.. “Harmony search algorithm for simultaneous clustering and feature selection”. Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of (pp. 202-207). IEEE. (2010)

[6] Revelt, D., & Train, K “Customer-specific taste parameters and mixed logit”. University of California, Berkeley. (1999).

Applying Fragments to the Dial a Ride Problem

Michael Forbes
School of Mathematics and Physics
The University of Queensland, St Lucia, Australia
Email: m.forbes@uq.edu.au

1 Introduction

In [1] a new exact method for the Pickup and Delivery Problem with Time Windows (PDPTW) was presented. This method is based on the idea of fragments - a series of pickup and delivery requests starting and ending with an empty vehicle. In this work we propose a novel exact method for the Dial a Ride Problem (DARP) by utilising similar techniques based on fragments. The DARP is a PDPTW in which the requests are people. Additional constraints are often imposed, most commonly maximum ride time. A comprehensive overview of DARP variations and solution techniques is given in [6]

In [1] the authors state that the "method is easy to implement and can be extended in a straightforward way to solve many variants of the PDPTW for problems where it is possible to generate all fragments." However, when their method is applied directly to the DARP it is not possible to generate all fragments. A key novelty of this work is to introduce the concept of "restricted fragments". A restricted fragment is a fragment which consists of a series of pickup requests followed by a series of delivery requests.

Using nodes and arcs derived from the restricted fragments, we build a relaxed network flow model with side constraints for coverage of all requests. We strengthen the relaxed formulation with a number of root node cuts and use lazy constraints to cut off any illegal solutions to the original problem found while solving the relaxed network flow model as an integer program. Computational results confirm that our method significantly outperforms state-of-the-art algorithms for solving the DARP.

2 Methodlogy

2.1 Restricted Fragments

Generating all admissible routes for even a modestly sized DARP instance is prohibitive. Branch-and-price approaches such as in [5] overcome this problem by implicitly considering all routes. Alyasiry et al. [1] instead propose using fragments. They define a fragment to be "part of legal vehicle route such that the vehicle starts empty at a pickup node and ends empty at a delivery node, but it is never empty at any intermediate node". Clearly any route can be represented as one or more fragment. For example, using upper case letters to represent pickup requests and lower case letters for the corresponding delivery requests, we may have a route corresponding to the series of requests: (ABaCcbDEFedGfg). This can be considered as two fragments connected end to end: (ABaCcb)+(DEFedGfg).

However for many problem instances even the number of fragments may grow too quickly. To address this problem, we define a restricted fragment as having all the characteristics of a fragment, but additionally restricted so that it only consists of a series of pickup requests followed by a series of delivery requests. If we consider the route above, relevant restricted fragments are (ABab), (BCcb), (DEFedf) and (FGfg). It will become clear later why these fragments model the example route.

2.2 Relaxed Network

After generating all restricted fragments we next proceed to build a relaxed network. We have two requirements for the relaxed network and the resultant network flow with side constraints formulation: any vehicle tour that is legal for the DARP must be a legal path in the relaxed network and it must be simple to add a constraint to cut off a path in the relaxed network that is not a legal DARP vehicle tour.

For a standard DARP with n requests we can designate the nodes as $N = \{0, 1, ..., n, n + 1, ..., 2n\}$ where 0 is the depot, i corresponds to the pickup node for request i and $i+n$ corresponds to the delivery node for request i.

In order to formulate our relaxed network, we define an "extended node" as the combination of a node and a partial load on a vehicle. An extended node can be written as (i, S) where $S \subset \{1, ..., n\}$ and $i \notin S$.

Our relaxed network is made up of: starting arcs that leave the depot and travel to all empty extended pickup nodes $(i, \emptyset), 1 \leq i \leq n$; ending arcs that return to the depot from all empty extended delivery nodes $(i, \emptyset), n+1 \leq i \leq 2n$; restricted fragment arcs that move from an extended pickup node to an extended delivery node; and repositioning arcs that move from an extended delivery node to an extended pickup node.

A restricted fragment arc is defined by the restricted fragment, starting extended node and ending extended node $[F,(p,P),(d,D)]$, where if $F=(i_1,i_2,...,i_{|F|})$ then $0 \leq |P| \leq |F|/2-1$, $0 \leq |D| \leq |F|/2-1$, $p=i_{|P|+1}$, $P=\{i_k, 1 \leq k \leq |P|\}$, $d=i_{|F|-|P|}$ and $D=\{i_k-n, |F|-|P|+1 \leq k \leq |F|\}$. An repositioning arc is defined by the starting extended node and ending extended node $[(d,D),(p,P)]$ where $D=P$ and $d \neq p$.

Our example route above can be considered as comprised of the following collection of arcs, where every second arc is a fragment arc: $[0,(A,\emptyset)]$ - $[(ABab),(A,\emptyset),(a,\{B\})]$ - $[(a,\{B\}),(C,\{B\})]$ - $[(BCcb),(C,\{B\}),(b,\emptyset)]$ - $[(b,\emptyset),(D,\emptyset)]$ - $[(DEFedf),(D,\emptyset),(d,\{F\})]$ - $[(d,\{F\}),(G,\{F\})]$ - $[FGfg,(G,\{F\}),(g,\emptyset)]$ - $[(g,\emptyset),0]$. What was initially one route, was first represented by five arcs (two depot movements, two fragments and one repositioning arc) and then by nine arcs (two depot movements, four restricted fragment arcs and three repositioning arcs).

2.3 Overall Algorithm

Our overall algorithm can be summarised as follows:

- Apply standard time window tightening and arc elimination rules as in [4].
- Generate all restricted fragments using a simple recursive procedure. We check maximum ride times using the procedure described in [5].
- Build the relaxed network. This is done carefully to exclude nodes and arcs which cannot possible be part of a feasible vehicle route.
- Build the integer programming model. This is a network flow model with side constraints to ensure each pickup node is covered and the number of vehicles is not exceeded. We also add constraints to eliminate cycles consisting solely of one fragment arc and one repositioning arc.
- Repeatedly solve the LP relaxation of the model and add cuts until no more cuts are added.
- Solve the IP, checking all integer solutions found and adding lazy constraints as required to cut off infeasible solutions. A solution polishing heuristic is also applied.

The cuts added to the LP relaxation are all of the form $x_a \leq \sum_{a' \in \Phi(a)} x_{a'}$, where x_a are variables corresponding to arcs and $\Phi(a)$ represents a set of arcs, one of which must be in use for arc a to be used. These constraints are effective because some combinations of in arcs and out arcs for a node are incompatible. For example, a fragment arc may not be compatible with every repositioning movement that leaves from its destination node.

3 Computational Results

The algorithm was run on all instances used in [5], including the extended examples where time windows and capacities were increased. These instances were originally presented in [2] and [3].

Our code was written in Python 3.6 using the callable library of Gurobi 8.0 to solve LP and MIP problems with four threads.

The table below shows the run times in seconds reported by [5] (the first column in each pair of columns, times limited to one hour) and our run times for some of the hardest instances, on broadly comparable hardware.

Instance	**Orig**		**4/3**		**5/3**		**6/3**	
b6-72	31.4	2.9	1691.4	9.9	3600.0	30.5	696.8	33.4
b7-56	50.3	2.2	26.8	6.8	38.5	23.7	347.4	67.0
b7-70	13.0	3.4	50.0	8.4	47.8	16.8	3600.0	189.5
b7-84	71.7	4.6	518.6	9.6	3600.0	77.9	3600.0	382.7
b8-64	23.1	1.9	9.3	8.9	73.5	18.5	3600.0	616.1
b8-80	10.3	1.0	15.4	6.7	3600.0	34.9	3600.0	670.2
b8-96	898.8	7.3	2135.6	19.7	3600.0	95.6	3600.0	4830.4

Table 1: Results on harder instances

The conference presentation will also report results on more difficult instances from other data sets as well as discussing possible extensions to the algorithm.

References

[1] A. Alyasiry, M. Forbes and M. Bulmer, "An Exact Algorithm for the Pickup and Delivery Problem with Time Windows and its Variants", Odysseus, Cagliari, June 2018.

[2] J-F. Cordeau, "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem", *Operations Research* 54, 573586 (2006).

[3] J-F. Cordeau and G. Laporte "The Dial-a-Ride Problem: Models and algorithms", *Annals of Operations Research* 153, 2946 (2007).

[4] Y. Dumas, J. Desrosiers and F. Soumis, "The Pickup and Delivery Problem with Time Windows", *European Journal of Operations Research* 54, 7-22 (1991).

[5] T. Gschwind and S. Irnich, "Effective Handling of Dynamic Time Windows and Its Application to Solving the Dial-a-Ride Problem", *Transportation Science* 49, 335-354 (2015).

[6] S. Ho, W. Szeto, Y. Kuo, J. Leung, M. Petering and T. Tou, "A survey of dial-a-ride problems: Literature review and recent developments", *Transportation Research Part B* 111, 395421 (2018).

Intermodal Rail Blocking and Car Fleet Management

Teodor Gabriel Crainic
Interuniversity Research Centre on Enterprise Networks,
Logistics and Transportation (CIRRELT) and
School of Management, Université du Québec à Montréal
Email: TeodorGabriel.Crainic@CIRRELT.net

Emma Frejinger
CIRRELT and Department of Computer Science and Operational Research,
Université de Montréal

Tien Mai
CIRRELT, Université de Montréal

Railroads are cost-effective and environmentally-sustainable. They move large quantities of a broad variety of commodities over long distances, and are thus a key element of the world-wide intermodal transportation network, displaying a steady traffic growth. Efficient and profitable railroad activities require adequate planning of operations and resources. These planning processes are complex due in large part to the interactions among the main components and goals of the system, e.g., yards, lines, trains, blocks of cars, economic profitability, resource utilization and customer satisfaction.

We focus on the tactical planning *Blocking & Car Fleet Management* problem (*BCFM*) for intermodal rail for which, according to our best knowledge, no adequate methodology exists. We propose a new *Scheduled Service Network Design with Resource Management* (*SSND-RM*) model for the BCFM that accounts for the characteristics of intermodality, namely, the demand expressed in terms of containers of various types and their assignment to multi-platform double-stack cars of different types. This container-to-car assignment requirement adds a new dimension to the train blocking problem addressed in the literature, and a new design layer to the classical car-to-block and block-to-train design decisions of the rail SND formulations. We briefly describe the problem (Section 1) and the model (Section 2), and sum up the presentation plan in Section 3.

1 Problem Description

Rail cargo is moved by trains made up of blocks of cars. Cars are classified (sorted) in yard terminals and assigned to blocks. A block is a group of cars, with possibly different origins and destinations, that move as a single unit between a pair of yards, without cars being handled individually when transferred from one train to another at intermediate yards. Blocking thus aims to take advantage of economies of scale and reduce yard handling costs. A block is moved by a sequence of trains, while a car can be moved by a sequence of blocks between its origin and destination yards. The classical *train blocking problem* consists in selecting the blocks to build and assigning cars to blocks. A number of studies in the literature address this case, e.g., [2, 3], but none accounts for the intermodal challenges: the need to explicitly account for the loading of containers on cars as well as the need to efficiently manage a limited fleet of cars.

About 90% of the containers used worldwide are 20 or 40 feet, while longer units, e.g., 53 feet, are also used in the North American market. The origin-to-destination (OD) demand is then defined as a number of containers of particular type and OD pair, as well as availability time at origin and due time at destination. This definition, in terms of cargo, is different from the classical one in terms of loaded cars. Moreover, the railroad uses a heterogeneous fleet of particularly-designed intermodal cars that, in most cases, is rented for the year.

These particular characteristics of the intermodal blocking problem induce two planning and methodological challenges. Consider, first, that each car has one to five platforms, which may be double- (two container slots) or single-stacked (one slot). The heterogeneity of the container fleet then yields a large number of loading alternatives and quite diverse numbers of required cars, even when considering the many existing loading rules [1]. Second, one cannot assume that the appropriate types and numbers of cars will always be available at all yards for all possible demands. Indeed, the car availability at a given yard and time instant is very much dependent on the earlier decisions regarding on what cars containers were loaded and how the cars where blocked.

Given a train schedule, the goal of the BCFM is to determine simultaneously a scheduled blocking plan that includes a block-to-car assignment policy and a car circulation to support the selected blocks, and that minimizes the total operation costs. Particularly challenging issues in addressing the BCFM are 1) simultaneously considering three consolidation processes, containers to cars, cars to blocks, and blocks to trains; 2) differentiating car and container types and representing in a computationally efficient way the assignment of containers to cars within a tactical blocking model; 3) integrating blocking and car-fleet management.

2 Modeling

We propose a model that is based on a cyclic four-layer space-time network representation, illustrated in Figure 1. This is a tactical problem and the plan is defined over a given *schedule length* (e.g., a week), to be repeated over the planning horizon. Since intermodal traffic shares the network

with trains moving other types of cargo, we take the train schedule as given. Different from most studies in the literature [3], this allows us to use a continuous-time network representation.

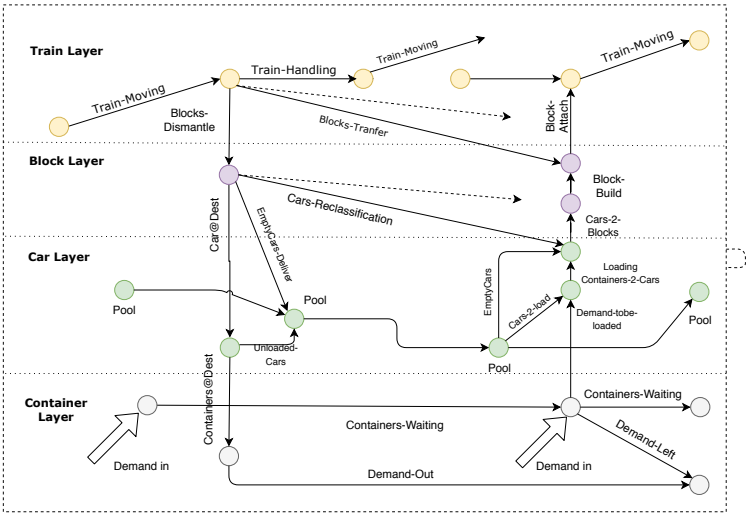


Figure 1: Four-Layer - Train, Block, car, Container - Time-Space Network Representation

The arrival and departure times of each intermodal *train* at the terminals on its route thus yield the corresponding arrival and departure nodes, defining the time structure of the entire network. We model train activities through *handling* arcs representing the time trains spend at terminals and *moving* arcs between terminals. The train features, e.g., power and maximum length, provide the capacity of the moving arcs. Each *block* in the potential-block set is defined by a particular time-dependent OD pair and a path made up of movements on train-moving arcs and activities in yards. We model *building* new blocks, *transferring* blocks from one train to another, and *dismantling* blocks at their destinations through appropriate intra-layer arcs connecting to the train and car layers.

The *car layer* provides the representation for 1) the container-to-car assignment and loading, and 2) the circulation of the car fleets. The latter takes place through car "pool" nodes and arcs standing for the inventory of empty cars (of the given type), together with arcs adding cars to (newly unloaded and empty delivered) and extracting cars from (to be loaded or shipped empty) the pool. The former take place on container-to-car loading arcs implementing the container-to-car assignment model we propose, based on the relations among the lengths of blocks, platforms (cars), and container types. Inter-layer arcs receive the loaded and empty cars at the destination of the block and send the loaded and empty cars to the selected block. Symmetrically, inter-layer arcs move the containers from / to the container layer when selected to be blocked and shipped or at destination, respectively. Finally, *demand* enters and exists the system through the *container layer*. Upon arrival, containers wait on *Container-Waiting* arcs until the selected block.

We propose a mixed integer linear programming (MILP) formulation with three groups of

decision variables: (i) *Block selection*, binary variables equal to 1 if a block is selected, and 0 otherwise; (ii) *Container flow distribution* representing the numbers of containers of appropriate demands on each block; (iii) *Car distribution* standing for the total number of cars of each type on each block, inventory arc, and loading/unloading arc.

The objective function minimizes the total cost of the system over the planning horizon. It encompasses the cost of selecting, operating and transferring blocks, the costs of handling and moving cars and containers, the time-related costs for containers and cars idling at train stops, as well as penalties for late arrival of demand and train-capacity overload. There are flow conservation constraints in the four layers and in between layers. Train and block capacities are enforced through linking constraints, and yard capacities are enforced as well. The latter are defined as bundle constraints regarding the maximum length of blocks that can be built and dismantled during a given time interval at a given terminal. Loading constraints yield the appropriate number of loaded cars given the assigned containers.

3 Conclusion

A large set of experiments was performed using data from a major North-American railroad using the model above as well as a more compact path-based formulation. Realistically-sized and defined instances were solved exactly within rather short computing times. We analyzed the impact on the design of the block & car plan, and the computational difficulty of the resulting instance, of several problem characteristics, e.g., number of trains and their number of intermediate stops, demand distribution in space and time, relative value of the penalties relative to the operational costs, possibility to split the demand flows among several blocks and the inclusion of extra trains (at high costs). We will present the problem, the container-to-car assignment solution we propose, the BCFM models, the experimental setting, and the numerical results and analyzes obtained.

We gratefully acknowledge the close collaboration with the Canadian National Railway Company (CN). This research is supported by the *CN Chair in Optimization of Railway Operations*, U. de Montréal, and the Natural Sciences and Engineering Research Council of Canada through its Collaborative Research & Development and Discovery grant programs.

References

[1] S. Mantovani, G. Morganti, N. Umang, T.G Crainic, E. Frejinger, and E. Larsen. The Load Planning Problem for Double-Stack Intermodal Trains. *EJOR*, 267(1):107–119, 2018.

[2] H.N Newton, C. Barnhart, and P.H Vance. Constructing Railroad Blocking Plans to Minimize Handling Costs. *Transportation Science*, 32(4):330–345, 1998.

[3] E. Zhu, T.G. Crainic, and M. Gendreau. Scheduled Service Network Design for Freight Rail Transportations. *Operations Research*, 62(2):383–400, 2014.

On the use of operations research methods for the design of school districts

Karen Smilowitz
Department of Industrial Engineering and Management Sciences
Northwestern University, Evanston, Illinois, United States
Email: ksmilowitz@northwestern.edu

1 Introduction

Operations research methodologies have been used to identify and evaluate solutions to the reconfiguration of public school attendance area boundaries for over fifty years. In broad terms, the school redistricting problem seeks to find capacity-feasible assignments of students in a school district to a local school (also referred to as "rezoning" or "optimizing attendance area boundaries"). Much of the early work was motivated by the movement to integrate schools. The years since have seen new directions of related work to address additional challenges related to the design of school attendance boundaries and leverage emerging advances in optimization and geographic information systems technology. Yet, many school districts still struggle with reconfiguring attendance area boundaries to meet changing needs of their communities. This analysis of the use of operations research for school districting is motivated by a collaboration with one such school district. As part of a larger research study looking at mathematical models to search for creative solutions to transportation challenges, the work to be presented is a reflection on the past fifty years of progress and the challenges remaining for school redistricting. While space prohibits a full literature review, including related problems such as political districting, the abstract highlights key related papers to provide context for this study. The presentation will feature a broad review of the literature, linking advances in the literature to current issues in education at the time and advances in technology.

2 Fifty years of research

While researchers have worked with school districts worldwide, many common themes have emerged over time, as researchers respond to changing factors in public education and the introduction

of new technology. In 1954, the United States Supreme Court ruled in Brown v. the Board of Education that segregating schools by race was unconstitutional and that schools should be integrated "with all deliberate speed". This ruling was followed by the 1968 Green v. New Kent ruling that called for significantly faster progress on integration. Around that time, papers began to appear in the operations research literature exploring analytical approaches to school integration. The titles of such papers reflect the focus on integration; e.g., "*School rezoning to achieve racial balance: a linear programming approach*" [8], "*An operations research approach to racial desegregation of school systems*" [5], and "*A network-flow model for racially balancing schools* [2]. As shown in the titles, these papers made use of advances in linear programming approaches of the time. Beginning with this early work and continuing today, assignment variables are typically defined by street segments or blocks where students live (referred to as tracts in early papers) and the schools to which students can be allocated. The above papers modeled the assignment variables as continuous variables, allowing split allocation of students at one location among different schools. Over time, it became more common to explicitly model the need to assign students on the same block or street segment to the same school through integer programming approaches, beginning with [10]. This is one example of the convergence in the growing ability to solve large-scale integer programs and evolving societal views on how students should be assigned to schools.

Above are three of many papers in the late 1960s and 1970s that applied emerging operations research techniques to school redistricting, identifying the unique features of the problem, particularly focusing on the trade-offs between achieving racial balance and minimizing travel distance for students. The dual consideration of achieving racial balance and minimizing travel distance naturally motivated new ways to communicate proposed solutions to decision makers. The 1990 work by Ferland and Guénette [7] is an early example of the move to improve visualization and interactive capabilities. Building on those ideas, a key feature in work over the past three decades has been the integration of optimization tools with Geographic Information Systems (GIS), recognizing the need to more seamlessly integrate geo-data with mathematical models of redistricting and the need to provide decision makers with more powerful visualization tools for interactive decision-making with school districts; see for example, [1] and [4].

Early papers considered the immediate need to integrate schools with greater speed; over time, school districts needed to again reconfigure attendance areas, even opening and closing schools, to meet changing demographics. For example, given declining enrollment in the 1970s and 1980s, many districts closed schools. Again, operations research models were developed to assist in the decision making. In 1975, [9] introduced integer variables to model the opening and closing of schools over time. School closings were considered a decade later in [6] which examines school decisions in a multi-objective framework, reflecting the complex mix of factors impacting school closing and reassignment decisions. While some models were designed for annual reconfiguration

of enrollments (e.g., [4]), many school districts anticipate a commitment of at least ten years for new configurations. However, modeling uncertainty in enrollment changes at a level suitable for redistricting models is challenging. Enrollment predictions for school districts are typically performed at more aggregated levels (e.g., the school or the entire district), whereas redistricting models consider more granular geographic units such as blocks or street segments. Armstrong et al. [1] addresses some of the challenges of including population projections in attendance area models.

Another challenge is the need to more explicitly consider the geography of the resulting boundary areas, including the compactness and contiguity of school neighborhoods. Such features are important in creating neighborhood cohesion and minimizing bus transportation. Compactness and contiguity are modeled in different ways in the political districting literature, but have received less attention (from a modeling perspective) in the school redistricting literature. Early work, including [9] and [6] among others, included objectives to minimize the sum of the squared distance from student location to school, which naturally results in compact regions. In [6], contiguity is considered through a set of constraints that ensure that a location in the district (in their case, defined as a cell in the region) is assigned to the same school as at least one neighbor. In [4], contiguity is modeled more directly through the use of internal paths connecting locations to schools. Increased computational power has been a factor in more explicit modeling of compactness and contiguity.

3 Implications for redistricting today

Fifty years after the work motivated by school integration, we continue to see new work in this area. Recent work has focused on economic integration of schools as research continues to show how income level impacts access to high quality education, see [3]. Trade-offs between providing equitable access to high quality schools and minimizing travel distances to schools are, in many ways, similar to the early conflicting objectives, but new challenges arise.

In the talk, we will present one such challenge. Beginning in 2015, Evanston / Skokie School District 65 partnered with researchers at Northwestern University to develop mathematical models to improve the service, efficiency and equity of transportation services. As others have done over the past fifty years, we are developing an interactive decision support tool to identify and evaluate potential changes to school attendance boundaries. School integration in Evanston, combined with school closings over time, has led to significant variation in the distances students travel to school and non-contiguous attendance area boundaries. We will present our work to incorporate the vast literature on the topic with new innovations to identify and evaluate changes to attendance area boundaries, ranging from incremental to longer-term, relative to key metrics that consider school

capacity levels, robustness to changing demographics, equity in access to schools, and potential reductions in bus transportation. Preliminary testing of the decision support tool is planned for February 2019 with the school district. At the conference, we will present observations from these tests. We will also discuss ways in which we are incorporating recent work on the approximation of covering path problems with network structure [11] into partitioning models to assist with the modeling of bus transportation costs.

References

[1] M. Armstrong, G. Rushton, R. Honey, "A spatial decision support system for school redistricting." *URISA Journal* 5 (1), 40-51 (1993).

[2] P. Belford and D. Ratliff, "A network-flow model for racially balancing schools", *Operations Research* 20 (3) 619-628 (1972).

[3] E. Bouzarth, R. Forrester, K. Hutson, and L. Reddoch, "Assigning students to schools to minimize both transportation costs and socioeconomic variation between schools", *Socio-Economic Planning Sciences* in press (2017).

[4] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub, "School redistricting: Embedding GIS tools with integer programming", *Journal of the Operational Research Society* 55 (8), 836-849 (2004).

[5] S. Clarke and J. Surkis, "An operations research approach to racial desegregation of school systems", *Socio-Economic Planning Sciences* 1 (3), 259-272 (1968).

[6] J. Diamond and J. Wright, "Multiobjective analysis of public school consolidation", *Journal of Urban Planning and Development*, 113(1), 1-18 (1987).

[7] J. Ferland and G. Gunette, "Decision support system for the school districting problem", *Operations Research* 38 (1), 15-21 (1990).

[8] L. Heckman and H. Taylor, "School rezoning to achieve racial balance: a linear programming approach", *Socio-Economic Planning Sciences* 3 (2), 127-133 (1969).

[9] C. Holloway, D. Wehrung, M. Zeitlin, and R. Nelson, "An interactive procedure for the school boundary problem with declining enrollment", *Operations Research*, 23(2), pp.191-206 (1975).

[10] R. Liggett, "The application of an implicit enumeration algorithm to the school desegregation problem", *Management Science*, 20 (2) 159-168 (1973).

[11] L. Zeng, S. Chopra, and K. Smilowitz, "The covering path problem on a grid", arXiv preprint arXiv:1709.07485 (2018).

A Continuous Model for Electric Vehicle Sharing with Battery Degradation

Jian Wu[1], **Xin Wang[1*]**, and **Feng Ju[2]**

[1]Department of Industrial and Systems Engineering

University of Wisconsin-Madison

[2] School of Computing, Informatics, and Decision Systems Engineering

Arizona State University

[*]**Corresponding author. Email: xin.wang@wisc.edu**

1 Introduction

Free-floating Electric Vehicle (EV) sharing are expected to be adopted on a tremendous scale due to its service flexibility and energy efficiency [1]. Despite EVs' great promise on the emission reduction, the extensive driving and charging of these vehicles could significantly reduce the life of batteries, or the state of health (SoH) [2]. Such inherent battery degradation process during long-term operations normally results in an underestimation of cost and implicitly affects the decision making of charging scheduling, dispatching, and deployment of the fleet. There is a lack of studies to address the critical impact of battery degradation in such EV sharing systems.

To fill this research gap, we propose an integrated model for EV sharing systems design in a continuous 2D space. This model aims to optimize the system design considering the joint impact of battery charging and degradation, hence provides decision support on the charge station deployment, fleet balancing operations, and the battery charging and replacing policy. An analytical solution is developed based on the Monge-Kantorovich problem, which provides fruitful managerial insights.

2 Preliminaries

Suppose a free-floating EV sharing system is operating on a continuous 2D service region Ω. Let $(x, y) \in \Omega \times \Omega$ be an origin-destination (OD) pair for a trip from location x to y, and the corresponding travel demand density nearby is $f(x, y)$ per unit time (service trip). Meanwhile, to balance the vehicles over the service region, the system operators should drive vehicles from

those places with idle vehicles, $\mathcal{X} \subset \Omega$, to those run out of vehicles, $\mathcal{Y} \subset \Omega$, to ensure vehicle availability (rebalance trip). Let the number of idle vehicles at $x \in \mathcal{X}$ per unit time be $\pi^+(x) = \int_\Omega [f(y,x) - f(x,y)]dy$, and the vehicle shortage at $y \in \mathcal{Y}$ be $\pi^-(y) = -\int_\Omega [f(x,y) - f(y,x)]dx$. Assume the energy consumptions for both service and rebalance trips are $d(x,y)$. The cost of hiring an operator to drive from x to y is $S(x,y)$ per unit time.

Each EV is characterized by a pair of State of Energy (SoE) and SoH (e,h). Let h_0 be the SoH of a brand new EV. The EV battery charging rate, which depends on both SoE and SoH, is denoted by $\kappa_e(e,h)$. In particular, we consider an impact factor $\lambda(h)$ such that $\kappa_e(e,h) = \lambda(h)\kappa_e(e,h_0)$. The degradation rate, which only dependents on SoH, is denoted by $\kappa_h(h)$. We also assume charge stations are densely installed in the service region. The unit power pump installation cost at x is $I(x)$. For simplicity, we assume the installation cost is homogeneous over Ω, i.e., $I(x) \equiv I_0$.

Now we describe the system decision. We denote the outbound service trip flow and rebalancing trip flow from x to y at a particular SoH-SoE level to be $\phi(x,y,e,h)$ and $\psi(x,y,e,h)$, respectively. Meanwhile, the corresponding inbound flows are denoted as $\phi'(x,y,e,h)$ and $\psi'(x,y,e,h)$. At location x, the number of EVs entering and exiting charge station per unit time is $\gamma(x,e,h)$ and $\gamma'(x,e,h)$, respectively. Suppose the operator retires a battery once its SoH reaches $\bar{h}$, and use a new battery (with SoH of h_0) at a cost C_0 as a replacement. Let $\Pi = \{(e,h)|0 \le e \le e_0(h), \bar{h} \le h \le h_0\}$ be the set of all feasible (e,h) pairs in system, where $e_0(h)$ is the maximum SoE level at at a given h. We have following EV flow balance equation:

$$\begin{aligned} &\int_\Omega \phi'(y,x,e,h)dy + \int_\Omega \psi'(y,x,e,h)dy + \gamma'(x,e,h) \\ &= \int_\Omega \phi(x,y,e,h)dy + \int_\Omega \psi(x,y,e,h)dy + \gamma(x,e,h), \quad \forall x \in \Omega, (e,h) \in \Pi \end{aligned} \tag{1a}$$

While charging, we model the change of battery SoE and SoH as a two-dimensional advection in Π. Let $n(x,e,h)$ be the density of EVs in the charge station at x with (e,h). The advection velocity is $\boldsymbol{\kappa} = (\kappa_e, -\kappa_h)$, indicating the SoE increases κ_e and SoH decreases κ_h per unit time. Then we have the following SoE-SoH advection equation in the charge station at x:

$$\nabla[\boldsymbol{\kappa}(e,h)n(x,e,h)] = \gamma(x,e,h) - \gamma'(x,e,h) \quad \forall (e,h) \in \Pi \tag{1b}$$

$$n(x,e,h)|_{h=h_0} = w_0(x,e) \qquad \text{(Boundary Condition)}$$

For location x, the rate of battery replacement with SoE of e is the product of w_0 and degradation rate $\kappa_h(h_0)$, denoted as $p(x,e)$. In addition, let $g(x,y)$ be the total rebalance flow from x to y.

Given the above problem setting, the system needs to find a policy to minimize the sum of charge station installation cost, vehicle rebalancing cost, and battery replacement cost.

Problem 2.1 ***(EV sharing system design problem (Continuous))***

$$\min_{\phi,\psi,\lambda,n,g \ge 0} \iiint_{\Omega\times\Pi} I_0 n(x,e,h)dedhdx + \int_\Omega \int_0^{e_0} C_0 p(x,e)dedx + \iint_{\Omega\times\Omega} S(x,y)g(x,y)dxdy \tag{1c}$$

s.t. Constraint (1a)(1b)

$$\iint_{\Pi} \phi(x,y,e,h)dedh = f(x,y), \iint_{\Pi} \psi(x,y,e,h)dedh = g(x,y) \qquad \forall x,y \in \Omega \quad (1d)$$

$$\int_{\mathcal{Y}} g(x,y)dy = \pi^{+}(x) \qquad \forall x \in \mathcal{X} \quad (1e)$$

$$\int_{\mathcal{X}} g(x,y)dx = \pi^{-}(y) \qquad \forall y \in \mathcal{Y} \quad (1f)$$

$$\phi'(x,y,e',h) = \phi(x,y,e,h), \psi'(x,y,e',h) = \psi(x,y,e,h) \qquad \forall e' = e - d(x,y) \geq 0 \quad (1g)$$

$$\phi(x,y,e,h), \psi(x,y,e,h) = 0 \qquad \forall e < d(x,y) \quad (1h)$$

3 Main results

Directly solving Problem 2.1 is extremely difficult. Note that, in most cases, the battery degradation rate is relatively neglectable compared to its charging rate, i.e. $\kappa_h \ll \kappa_e$. Hence the changes of EVs' SoH can be ignored in a single charging cycle. To this end, the two-dimension advection can be decomposed into two one-dimension advection in Π.

$$\frac{\partial}{\partial e}[\kappa_e(e,h)n(x,e,h)] = \int_{\Omega} [\phi'(y,x,e,h) + \psi'(y,x,e,h) - \phi(x,y,e,h) - \psi(x,y,e,h)]dy \qquad (2a)$$

$$\frac{\partial}{\partial h}[-\kappa_h(h)n(x,e,h)] = 0 \qquad (2b)$$

$$n(x,e,h)|_{h=h_0} = w_0(x,e) \qquad \text{(Boundary Condition)}$$

Considering that each trip for (x,y) is infinitesimal in a continuous space, we can assume an operating policy such that an infinitesimal outbound flow on (x,y) share the same SoE level. Following such operation, we denote the outbound SoE for service and rebalance trips to be $\mu(x,y)$ and $\nu(x,y)$, respectively. Thus when these EVs arrive destination, their SoE become $\mu'(x,y) = \mu(x,y) - d(x,y)$ and $\nu'(x,y) = \nu(x,y) - d(x,y)$. Integrating both sides of (2a) on $\Omega \times \Pi$ and applying simple algebra, we can get

$$\iiint \lambda(h)n(x,e,h)dxdedh = \iint (\int_{\mu'(x,y)}^{\mu(x,y)} \frac{f(x,y)}{\kappa_e(e,h_0)}de + \int_{\nu'(x,y)}^{\nu(x,y)} \frac{g(x,y)}{\kappa_e(e,h_0)}de)dxdy \qquad (3)$$

We denote $\tilde{\kappa}_h(\bar{h}) = (\int_{\bar{h}}^{h_0} \kappa_h^{-1}dh)^{-1}$. Note $\tilde{\kappa}_h(\bar{h})$ is the effective battery degradation rate when the retiring SoH is $\bar{h}$. Denote $\tilde{\kappa}_e(e,\bar{h}) = \int_{\bar{h}}^{h_0} \kappa_e/\kappa_h dh \cdot \tilde{\kappa}_h(\bar{h})$. Here, $\tilde{\kappa}_e(e,\bar{h})$ is the effective charging rate. Note we can extract $\kappa_e(e,h_0)$ from $\tilde{\kappa}_e(e,\bar{h})$, which yields $\tilde{\lambda}(\bar{h}) = \int_{\bar{h}}^{h_0} \lambda/\kappa_h dh \cdot \tilde{\kappa}_h(\bar{h})$. We can get the total number of charging pumps N and w_0 by Equation (2)(3), and get Problem 3.1.

$$N = \iint [\int_{\mu'(x,y)}^{\mu(x,y)} \frac{f(x,y)}{\tilde{\kappa}_e(e,\bar{h})}de + \int_{\nu'(x,y)}^{\nu(x,y)} \frac{g(x,y)}{\tilde{\kappa}_e(e,\bar{h})}de]dxdy = \frac{\kappa_h(h_0)}{\tilde{\kappa}_h(\bar{h})} \iint w_0(x,e)dedx$$

Problem 3.1 ***(EV sharing system design problem (slow degradation))***

$$\min_{g,\mu,\mu',\nu,\nu',\bar{h} \geq 0} (I_0 + \frac{C_0}{\tilde{\kappa}_h(\bar{h})}) \cdot \iint_{\Omega \times \Omega} f(x,y)K(\mu',\mu) + g(x,y)[K(\nu',\nu) + S(x,y)]dxdy \qquad (4a)$$

s.t. Constraint (1e)(1f)

$$\mu(x,y), \nu(x,y) \leq e_0(\bar{h}) \qquad \forall x, y \in \Omega \tag{4b}$$

where $K(\mu', \mu) = \int_{\mu'(x,y)}^{\mu(x,y)} [\tilde{\kappa}_e(e, \bar{h})]^{-1} de$.

Theorem 3.1 *If the charging rate κ_e is a monotone decreasing function over e, the optimal solution of μ, ν, and $\bar{h}$ is given as (5)(6). Moreover, if both $d(x,y)$ and $S(x,y)$ are proportional to $\|x-y\|$, Problem 3.1 can be reduced to the Monge-Kantorovich (M-K) problem with a strictly convex transport cost.*

$$\mu^*(x,y) = \nu^*(x,y) = d(x,y) \tag{5}$$

$$\bar{h}^* = \underset{\bar{h}}{\operatorname{argmin}} \ \frac{I_0 + C_0/\tilde{\kappa}_h(\bar{h})}{\tilde{\lambda}(\bar{h})} \tag{6}$$

Equation (5) tells EVs should operate at the lowest possible energy level as long as the energy usage requirement is met. Equation (6) implies the optimal retiring SoH value minimizes an equivalent infrastructure cost considering battery retiring and charging rate slow down. The numerator is the sum of charging pump cost and battery life degradation cost, while the denominator serves as an adjustment factor for charging efficiency due to battery degradation.

Finally, to solve the optimal rebalancing policy $g(x,y)$, we first define a transport cost $c(x,y) := K(\nu'^*, \nu^*) + S(x,y)$ [3]. Then the Kantorovich potential u (similar to node potential of min-cost flow problem), which is the dual price of the flow balance constraint at certain location, can be calculated as $u(x) = \min_{y \in \mathcal{Y}}(u(y) + c(x,y)), \forall x \in \mathcal{X}$ and $u(y) = \max_{x \in \mathcal{X}}(u(x) - c(x,y)), \forall y \in \mathcal{Y}$. The unique optimal rebalancing map $s(x)$ satisfies $u(x) - u(s(x)) = c(x, s(x))$. Thus $g(x,y)$ satisfies $g(x, s(x)) = \pi^+(x), \forall x \in \mathcal{X}$, which implies all rebalance trips from x should target at an optimal location $s(x)$.

4 Conclusion

We establish a novel framework to study the optimal design problem of free-floating EV car sharing system considering charging, rebalancing, and battery replacement operations. The problem deals with a heterogeneous fleet with different battery SoE and SoH, nonlinear charging and degradation rate, and a service region with densely installed charge stations. We formulate a model in continuous space and prove that the model can be reformulated as an optimal transport problem. An analytical solution is obtained under the assumption that the degradation process is slow. Our approach provides managerial insights for such EV sharing services providers in terms of charge station deployment, EV dispatching, rebalancing operations, and retiring policies.

References

[1] L. He, H.Y. Mak, Y. Rong, and Z. J. M. Shen, "Service region design for urban electric vehicle sharing systems", *Manufacturing and Service Operations Management* 19(2), 309-327 (2017).

[2] S. B. Peterson, J. Apt, and J. F. Whitacre, "Lithium-ion battery cell degradation resulting from realistic vehicle and vehicle-to-grid utilization", *Journal of Power Sources* 195(8), 2385-2392 (2010).

[3] L. Caffarelli, M. Feldman, and R. McCann, "Constructing optimal maps for Monges transport problem as a limit of strictly convex costs", *Journal of the American Mathematical Society* 15(1), 1-26 (2002).

Appendix: Proofs

4.1 Equation (3)

$$\frac{\partial}{\partial e}[\kappa_e(e,h)n(x,e,h)] = \int_\Omega [\phi'(y,x,e,h) + \psi'(y,x,e,h) - \phi(x,y,e,h) - \psi(x,y,e,h)]dy \tag{7}$$

$$\kappa_e(e,h)n(x,e,h) = \int_\Omega [\Phi'(y,x,e,h) + \Psi'(y,x,e,h) - \Phi(x,y,e,h) - \Psi(x,y,e,h)]dy \quad \text{(cumulative on e)} \tag{8}$$

$$\lambda(h)n(x,e,h) = \frac{1}{\kappa_e(e,h_0)} \int_\Omega [\Phi'(y,x,e,h) + \Psi'(y,x,e,h) - \Phi(x,y,e,h) - \Psi(x,y,e,h)]dy \tag{9}$$

$$\iint \lambda(h)n(x,e,h)dxde = \iint (\int_{\mu'(x,y)}^{\mu(x,y)} \frac{f_h(x,y)}{\kappa_e(e,h_0)}de + \int_{\nu'(x,y)}^{\nu(x,y)} \frac{g_h(x,y)}{\kappa_e(e,h_0)}de)dxdy \tag{10}$$

$$\iiint \lambda(h)n(x,e,h)dxdedh = \iint (\int_{\mu'(x,y)}^{\mu(x,y)} \frac{f(x,y)}{\kappa_e(e,h_0)}de + \int_{\nu'(x,y)}^{\nu(x,y)} \frac{g(x,y)}{\kappa_e(e,h_0)}de)dxdy \tag{11}$$

4.2 N and w_0

First solve w_0

$$\frac{\partial}{\partial h}[-\kappa_h(h)n(x,e,h)] = 0 \quad and \quad n(x,e,h)|_{h=h_0} = w_0(x,e) \tag{12}$$

$$\kappa_h(h)n(x,e,h) = \kappa_h(h_0)w_0(x,e) \tag{13}$$

$$\iint n(x,e,h)dxde = \frac{1}{\kappa_h(h)}\kappa_h(h_0) \iint w_0(x,e)dedx \tag{14}$$

$$N = \int \frac{1}{\kappa_h(h)}dh \cdot \kappa_h(h_0) \iint w_0(x,e)dedx \quad \text{(Integrate on h)} \tag{15}$$

$$N = \frac{\kappa_h(h_0)}{\tilde{\kappa}_h(\bar{h})} \iint w_0(x,e)dedx \tag{16}$$

From above Equation (11), we have

$$\int \lambda(h)[\iint n(x,e,h)dxde]dh = \iint (\int_{\mu'(x,y)}^{\mu(x,y)} \frac{f(x,y)}{\kappa_e(e,h_0)}de + \int_{\nu'(x,y)}^{\nu(x,y)} \frac{g(x,y)}{\kappa_e(e,h_0)}de)dxdy := \mathcal{P} \tag{17}$$

Plug in (14) to (17)

$$\int [\frac{\lambda(h)}{\kappa_h(h)} \cdot \kappa_h(h_0) \iint w_0(x,e) dedx] dh = \mathcal{P} \tag{18}$$

$$\int \frac{\lambda(h)}{\kappa_h(h)} dh \cdot \kappa_h(h_0) \iint w_0(x,e) dedx = \mathcal{P} \tag{19}$$

$$\kappa_h(h_0) \iint w_0(x,e) dedx = \frac{1}{\int \lambda(h)/\kappa_h(h) dh} \mathcal{P} \tag{20}$$

Compare (20) with (14)

$$\iint n(x,e,h) dxde = \frac{1/\kappa_h}{\int \lambda(h)/\kappa_h(h) dh} \mathcal{P} \tag{21}$$

$$N = \frac{\int 1/\kappa_h dh}{\int \lambda(h)/\kappa_h(h) dh} \mathcal{P} = \frac{1}{\tilde{\lambda}(\bar{h})} \mathcal{P} \quad \text{(Integrate on h)} \tag{22}$$

A new Benders decomposition method for metropolitan container logistics problems

Andrew Perrykkad[1], Andreas Ernst[1], and Mohan Krishnamoorthy[2]

[1]*School of Mathematical Sciences, Monash University, Australia*
[2]*School of Information Technology and Electrical Engineering, The University of Queensland, Australia*

One of the key bottlenecks of the international container supply chain is the transportation of containers between deep-sea container ports and the surrounding hinterland area. This transportation task is predominantly carried out by trucks, however recently various multi-modal approaches have begun to appear—employing road transportation in conjunction with rail or other transport modalities. Under these configurations, containers may be transported directly to the port by truck or instead to an *inland container terminal*, with the remainder of the journey facilitated by a short-haul rail service. Within the operations research literature, the optimisation of these highly-complex hinterland container transportation networks to ensure all orders are transported punctually whilst minimising cost is referred to as the *hinterland* or *inland container transportation problem*.

A number of authors have considered variations on the inland container transportation problem. At the tactical level, Li, Negenborn, and De Schutter [1] develop a flow-based model for a transportation network consisting of rail, road, and inland waterway (barge) transportation and use a rolling-horizon control to optimise freight movements around the Port of Rotterdam. Zhang and Pel [5] additionally consider a case-study for the Port of Rotterdam hinterland, also developing a flow-based model and assessing various competing transportation paradigms. At a more operational level, Wang and Yun [4] consider an intermodal freight transportation problem with a single rail link and associated drayage activities—the authors develop a hybrid Tabu search heuristic and evaluate it over randomly generated instances. A comprehensive review of problems in this space is given by SteadieSeifi, Dellaert, Nuijten, Van Woensel, and Raoufi [3].

Within Europe (where most previous research has been done) and the USA, hinterland areas are commonly large and diffuse. This contrasts dramatically with the Australian context, in which there exist much more compact and well-defined metropolitan areas around ports. We consider a variant of the inland container transportation problem arising from this context and refer to it as the *metropolitan freight transportation problem with single allocated drayage* (MFTP). Our key aim is to understand the link between rail service allocation and road transportation and as such in this paper consider the special case with unit demand and unit truck capacity (MFTP-1AD).

Let Q be a set of *orders*—partitioned into a set of imports D and exports P—to be satisfied within the planning horizon. Import orders must be transported from the port to their final import destination, and export orders from their export origin to the port. All orders are assumed to consist of one *twenty-foot equivalent unit* (TEU) and must be satisfied within the planning horizon. Both road and rail modalities are available to transport containers. We denote S as the set of rail terminals (and equivalently rail lines) within the network, with the port included in this set of terminals. Transportation of containers along a rail line s is charged at α_s^D per TEU for imports and α_s^P per TEU for exports. Similarly, each rail line has an outbound capacity of h_s^D TEUs and an inbound capacity of h_s^P TEUs.

Additionally, the road network is administered by a homogeneous fleet of vehicles, each with unit capacity; restricting road routes to routes of length two (out-and-back for an import or export) or length three (an import order paired with an export order). We assume that for an import and export order to be paired they must both pass through the same container terminal. The cost of moving between locations $a \in Q \cup S$ and $b \in Q \cup S$ is given by γ_{ab} (not necessarily symmetric). A visualisation of an example instance is given as Figure 1.

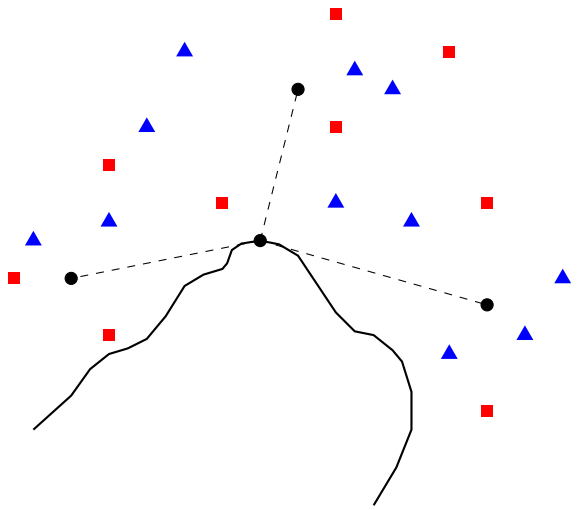


Figure 1: Geographical distribution of an example metropolitan area. Here blue triangles (▲) represent import locations, red squares (■) represent export locations, and black circles (●) represent terminal locations. Dashed black lines represent rail lines between terminals and the port (central circle) and the solid black curve the coastline. The MFTP seeks to allocated orders to container terminals, then subsequently link import and export orders together in a single road route.

Under the MFTP-1AD, we seek to satisfy all orders in Q at minimum cost whilst respecting all capacity constraints. Consider an integer programming formulation for the MFTP-1AD on the binary variables:

- y_{qs}, indicating whether customer request q is transported on rail line s.
- x_{ijs}, indicating that import request i and export request j are satisfied in a single route associated with rail terminal s (i.e. the truck travels along path (s, i, j, s)). A direct road service—(s, i, s) or (s, j, s)—is modelled by the dummy index b in place of i or j, respectively.

Given these decision variables, the goal is to minimise the function

$$F(\mathbf{x}, \mathbf{y}) = \sum_{s \in S} \left(\sum_{i \in D} (\alpha_s^D + \gamma_{si}) y_{is} + \sum_{j \in P} (\alpha_s^P + \gamma_{js}) y_{js} + \sum_{i \in D} \sum_{j \in P} \gamma_{ij} x_{ijs} + \sum_{i \in D} \gamma_{is} x_{ibs} + \sum_{j \in P} \gamma_{sj} x_{bjs} \right), \tag{1}$$

i.e. the total transportation cost as the sum of the rail transportation, drayage, and direct and paired dead-heading costs, whilst respecting all constraints of the system. The integer program (2)–(8) then defines an optimal solution to the MFTP-1AD:

$$\min_{x,y} : \quad F(\mathbf{x}, \mathbf{y}) \tag{2}$$

$$\text{s.t.} \quad \sum_{s \in S} y_{qs} = 1, \quad \forall q \in Q \tag{3}$$

$$\sum_{i \in D} y_{is} \leq h_s^D, \quad \forall s \in S \tag{4}$$

$$\sum_{j \in P} y_{is} \leq h_s^P, \quad \forall s \in S \tag{5}$$

$$\sum_{j \in P \cup \{b\}} x_{ijs} = y_{is}, \quad \forall i \in D, s \in S \tag{6}$$

$$\sum_{i \in D \cup \{b\}} x_{ijs} = y_{js}, \quad \forall j \in P, s \in S \tag{7}$$

$$y_{qs}, x_{ijs} \in \{0, 1\}, \quad \forall s \in S, i \in D \cup \{b\}, j \in P \cup \{b\}, i \neq j, \tag{8}$$

Here, (2) minimises the cost function (1). Equation (3) ensures each order is allocated to exactly one rail

terminal. Constraints (4) and (5) ensure that rail service capacity restrictions are enforced. Equations (6) and (7) ensure that an order may only be serviced by road from a terminal if it is allocated to that terminal. These constraints also ensure that each order appears as part of exactly one road route. Finally, equation (8) provides the domain for variable sets x and y.

In practice, solving instances of the MFTP-1AD with traditional MIP techniques can prove challenging as the problem is NP-hard. We demonstrate this for the case where vehicle capacity forms part of the input and for any fixed integer vehicle capacity.

Theorem 1 *The MFTP is NP-hard if truck capacity $g \in \mathbb{Z}^+$ is part of the input.*

Theorem 2 *The MFTP is NP-hard for all fixed truck capacities $g \in \mathbb{Z}^+$.*

To solve the problem more effectively, we develop a Benders decomposition with rail-allocation decisions made in the master problem (MP) and road transportation decisions in the subproblem (SP). As there always exists a feasible road transportation strategy for a given rail-assignment, only Benders optimality cuts are required. Additionally, due to the requirement that import and export orders are only 'paired' if they are associated with the same terminal, the subproblems may be completely decoupled. More formally, in each iteration the rail allocation variables y_{qs} are fixed to some $\hat{y}_{qs}$ for $q \in Q, s \in S$. Each iteration we therefore obtain $|S|$ independent integer programming subproblems defined by the vector $\hat{\mathbf{y}} = (\hat{y}_{qs})$:
($\mathrm{SP}_s(\hat{\mathbf{y}})$)

$$\min_{x}: \quad \sum_{i\in D}\sum_{j\in P}\gamma_{ij}x_{ijs} + \sum_{i\in D}\gamma_{is}x_{ibs} + \sum_{j\in P}\gamma_{sj}x_{bjs} \tag{9}$$

$$\sum_{j\in P\cup\{b\}} x_{ijs} = \hat{y}_{is}, \quad \forall i \in D \tag{10}$$

$$\sum_{i\in D\cup\{b\}} x_{ijs} = \hat{y}_{js}, \quad \forall j \in P \tag{11}$$

$$x_{ijs} \in \{0,1\}, \quad \forall i \in D\cup\{b\}, j \in P\cup\{b\}, i \neq j, \tag{12}$$

In its current form, the mathematical program (9)–(12) cannot be dualised (and as such cannot be used to generate Benders cuts) due to the integrality constraints (12). Fortunately, the subproblem may be solved equivalently as a minimum-cost flow problem—allowing us to relax the integrality constraint due to the total unimodularity property of network flow problems, in addition to allowing the use of specialised network solution algorithms.

Due in part to degeneracy in dual subproblem, convergence of the Benders decomposition proved underwhelming, and as such we developed a Magnanti-Wong acceleration for our Benders decomposition algorithm. Under their methodology Magnanti and Wong [2] generate pareto-optimal Benders cuts by solving a second linear programming problem related to the dual subproblem. We refer to this second LP as the *Magnanti-Wong dual subproblem*. The Magnanti-Wong dual subproblem chooses from optimal solutions to the dual subproblem by adding a constraint to the LP fixing the objective expression to the optimal objective value. The Magnanti-Wong subproblem them optimises with respect to *core-point*—an element of the relative interior of the convex hull of the master problem (here denoted $\mathbf{m} \in \mathbb{R}^{|Q|\times|S|}$).

Under our Benders decomposition, taking the dual of the Magnanti-Wong dual subproblem gives the Magnanti-Wong primal problem: ($\mathrm{MWP}_s(\hat{\mathbf{y}})$)

$$\min_{x,z}: \quad \sum_{i\in D}\sum_{j\in P}\gamma_{ij}x_{ijs} + \sum_{i\in D}\gamma_{is}x_{ibs} + \sum_{j\in P}\gamma_{sj}x_{bjs} - \hat{\delta}_s z_s \tag{13}$$

$$\sum_{j\in P\cup\{b\}} x_{ijs} - \hat{y}_{is}z_s = m_{is}, \quad \forall i \in D \tag{14}$$

$$-\sum_{i\in D\cup\{b\}} x_{ijs} + \hat{y}_{js}z_s = -m_{js}, \quad \forall j \in P \tag{15}$$

$$x_{ijs}, z_s \geq 0, \quad \forall i \in D\cup\{b\}, j \in P\cup\{b\}, i \neq j, \tag{16}$$

where $\hat{\delta}_s$ is the optimal objective value of subproblem $\mathrm{SP}_s(\hat{\mathbf{y}})$ and z_s the new 'primal' variable associated with the Magnanti-Wong fixing constraint in the dual.

Although providing superior convergence to the out-of-the-box Benders implementation, our Magnanti-Wong acceleration still suffers from two problems persistent with the method: the need for two LP solves per iteration and the destruction of any network structure in the (second) LP subproblem. By noting that it is possible to find an optimal solution to $\text{MWP}_s(\hat{\mathbf{y}})$ without the value of $\hat{\delta}_s$ (Lemma 3) we developed the *simultaneous Magnanti-Wong method* (Theorem 4): requiring just one LP solve per Benders iteration and preserving the underlying network structure.

Lemma 3 *Let Q_s be the subset of orders Q allocated (at least in part) to terminal $s \in S$ under the current master problem solution,*

$$Q_s \coloneqq \{q \in Q \mid y_{qs} > 0\},$$

and let $\hat{y}_{bs}$ be the 'export deficit' for the current master problem solution,

$$\hat{y}_{bs} \coloneqq \sum_{j \in P} \hat{y}_{js} - \sum_{i \in D} \hat{y}_{is}.$$

Now define the set Q'_s as $Q_s \cup \{b\}$ if $\hat{y}_{bs}$ is nonzero and Q_s otherwise. Lastly define the constant M_s,

$$M_s \coloneqq \max\left\{\sum_{i \in D} m_{is}, \sum_{j \in P} m_{js}\right\}.$$

Then for non-negative $\mathbf{m}$, there exists an optimal solution to $MWP_s(\hat{\mathbf{y}})$ with $z_s = \hat{z}$, for all $\hat{z} \geq L_s$, where

$$L_s \coloneqq \max_{q \in Q'_s}\left\{\frac{M_s}{|\hat{y}_q|}\right\}.$$

Theorem 4 *For non-negative $\mathbf{m}$, the linear program generated by fixing variable z_s to any $\hat{z}_s \geq L_s$: $(sMWP_s(\mathbf{m}, \hat{\mathbf{y}}))$*

$$\begin{aligned}
\min_x: \quad & \sum_{i \in D}\sum_{j \in P} \gamma_{ij} x_{ijs} + \sum_{i \in D} \gamma_{is} x_{ibs} + \sum_{j \in P} \gamma_{sj} x_{bjs} \\
& \sum_{j \in P \cup \{b\}} x_{ijs} = m_{is} + \hat{y}_{is}\hat{z}_s, \quad \forall i \in D && (17) \\
- & \sum_{i \in D \cup \{b\}} x_{ijs} = -m_{js} - \hat{y}_{js}\hat{z}_s, \quad \forall j \in P && (18) \\
& x_{ijs} \geq 0, \quad \forall i \in D \cup \{b\}, j \in P \cup \{b\}, i \neq j
\end{aligned}$$

defines a minimum-cost flow problem. If $\mathbf{m}$ is a core point of MP, an optimal dual solution $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$, where $\mathbf{u}$ are the dual variables for constraints (17) *and $\mathbf{v}$ the dual variables for constraints* (18)*, defines a pareto-optimal Benders optimality cut for the MFTP-1AD.*

Computational experiments were run to evaluate the performance of the simultaneous Magnanti-Wong method against other variations of the algorithm (the out-of-the-box Benders decomposition and basic Magnanti-Wong acceleration) in addition to the standard MIP implementation, across simulated and real world instances, demonstrating significant performance improvements via our method.

References

[1] L. Li, R. R. Negenborn, and B. De Schutter. Intermodal freight transport planning – a receding horizon control approach. *Transportation Research Part C: Emerging Technologies*, 60:77–95, 2015.

[2] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[3] M. SteadieSeifi, N. P. Dellaert, W. Nuijten, T. Van Woensel, and R. Raoufi. Multimodal freight transportation planning: A literature review. *European Journal of Operational Research*, 233(1): 1–15, 2014.

[4] W. F. Wang and W. Y. Yun. Scheduling for inland container truck and train transportation. *International Journal of Production Economics*, 143(2):349–356, 2013.

[5] M. Zhang and A. J. Pel. Synchromodal hinterland freight transport: Model study for the port of rotterdam. *Journal of Transport Geography*, 52:1–10, 2016.

Solving The Joint Multi-School Bell Time and Route Scheduling Optimization Problem

Ali Haghani

Department of Civil and Environmental Engineering, University of Maryland - College Park, MD 20742, USA, Email: haghani@umd.edu

Ali Shafahi

Department of Computer Science, University of Maryland - College Park, MD 20742, USA, Email: ashafahi@umd.edu

Zhongxiang Wang*

Department of Civil and Environmental Engineering, University of Maryland - College Park, MD 20742, USA, Email: zxwang25@umd.edu

1 Introduction

The school bus transportation planning problem (SBPP) has been well studied. Due to its computational complexity, many papers decomposed it into several subproblems. The most widely used decomposition method was proposed by Desrosiers et al. [1], which decomposed the SBPP into four subproblems: bus stop selection (SS), bus trip generation (TG), school bell time adjustment (TA) and route scheduling (RS). The bus stop selection (SS) finds a set of bus stops such that the students can walk to these stops and take the school buses. The trip generation (TG) constructs a collection of school-exclusive trips where each trip is an ordered sequence of bus stops ending at the school (for morning trip) or starting at the school (for afternoon trip). The bell time adjustment (TA) find the best school bell times (or dismissal time) within a certain time window such that a good route scheduling plan can be obtained. The route scheduling (RS) groups compatible trips (from different schools) into blocks and serves each block with one bus. An ordered trip pair is compatible if the end time of the first trip plus the deadhead from the last stop of the first trip to the initial stop of the second trip is less than or equal to the start time of the second trip.

Many efforts have been focusing on the bus stop selection, trip generation, and route scheduling while the bell time adjustment is severely neglected. Its importance is without question. Our empirical cooperation with several public school districts in Maryland and Colorado indicated that the school board is more willing to change the school bell time than the bus stop locations and the trips. The latter two would have a bigger impact on the students, parents, teachers, bus drivers, infrastructure and the whole transportation system than the bell times.

The bell time adjustment problem is usually incorporated into the school bus scheduling problem [2]. Fügenschuh [2] proposed a trip-based Mixed Integer Linear Programming (MIP) model to

optimize the bell time adjustment and routing scheduling problem. The problem is solved using commercial solver with LP (linear programming) relaxation strengthen techniques like start time propagation, variable fixing, big-M reduction, coefficient lifting, cutting planes. The method can solve problem up to 102 schools and 490 trips. Then, Fügenschuh [3] developed a set partitioning reformulation method to solve the same problem with two cutting planes: set-cover inequality and clique inequality. However, both methods are lack of ability to solve large-scale real-world problems. In this paper, we present a novel and more efficient MIP model and a local search-based heuristic algorithm to solve the Joint Multi-School Bell Time Adjustment and Route Scheduling Optimization Problem (TARS).

2 Methodology

Fügenschuh [2] and Fügenschuh [3] both require that all trips arrive at school within 5 minutes of the school bell times. A more common practice is that all afternoon trips from one school depart at the school dismissal time[1]. It helps to avoid the safety issues of holding some students after school because of the different departure time of the trips. Under this assumption, we present a new School-based Bell Time Adjustment and Route Scheduling (S-TARS) model, which is much more efficient than the trip-based formulation from Fügenschuh [2]. The comparison of the two models is shown in Table 1, including the analytical analysis and a real-world problem from the Howard County Public School System (HCPSS) in Maryland with 78 schools and 994 trips.

Table 1 Comparison of two MIP models

Model		Fügenschuh [2]		S-TARS	
Objective		NOB+DD		NOB+DD	
Constraints		1. Trip compatibility 2. School Bell Time Window 3. Trip departs within 5 mins of its school's bell time 4. The trip sequence on each bus		1. Trip compatibility 2. School Bell Time Window 3. The trip sequence on each bus	
Assumption		1. Maximum idle time 2. Trips depart within 5 mins of its school's bell time		1. No limit on idle time 2. Trips depart at the school dismissal time	
Comparison		Analytical	HCPSS	Analytical	HCPSS
# of variables	Binary	$\|E\| + 2\|N\|$	974,708	$\|P\| + 2\|N\|$	78,526
	Integer	$\|M\|$	78	$\|M\|$	78
	Continuous	$\|N\|$	994	-	0
# of constraints		$4\|E\| + 5\|N\| + 2\|M\| + \|P\|$	3,972,544	$2\|N\| + \|M\| + \|P\|$	78,604

Note: **Objective:** NOB: number of buses; DD: deadhead duration; **Sets**: M: set of schools; M_t: a school that trip t belongs to; N: a set of trips, $|M| \leq |N|$; N_m: set of trips for school m; $E = \{t1, t2\}\ \forall t1, t2 \in N | M_{t1} \neq M_{t2}, |E| \leq |N|^2$; $P = \{t1, s2\} \forall t1 \in N, s2 \in M | t1 \notin N_{s2}, |P| \leq |M| \times |N| \leq |N|^2$.

The result shows that the new school-based formulation (S-TARS) reduced 91% of the binary variables and 98% of constraints from Fügenschuh [2]'s formulation. Such a huge improvement mainly comes two simplifications:

[1] The morning and afternoon problems are identical, since one can easily be solved by reversing the solution to the other one. We solved the afternoon problem.

1) In S-TARS, all trips depart at the school dismissal time as a contrast to a more relaxed assumption that trips depart within 5 minutes of school bell time in Fügenschuh [2];
2) The maximum idle time constraint in Fügenschuh [2] is relaxed in S-TARS.

Due to the first simplification, the trip-to-trip variable (E) can be replaced by the trip-to-school variable (P) and the latter is much smaller than the former ($|P| \ll |E|$). In HCPSS, $|P| = 76{,}538$ and $|E| = 972{,}720$. Second, the maximum idle time, which is the time difference between a bus arrives at a school and the actual departure time of an afternoon trip, is relaxed in S-TARS. Such constriant is less important and is easy to implement in practice: the bus goes back and waits at the bus yard if the idle time is too long. Thanks to these two reasonable simplifications, the problem is significantly reduced using the school-based formulation (S-TARS).

The S-TARS is shown to be effective in solving the small to the medium-sized problem in a much shorter time than Fügenschuh [2]. However, it is still a little bit slow to solve a large-scale real-world problem like HCPSS. Thus, a local search-based heuristic algorithm is proposed. The basic idea of this algorithm is that given a solution, it fixes some schools' current dismissal times and finds its best neighbor solution by optimizing the dismissal time and route schedule for the free schools. The number of fixed and free schools along with the choice of these schools are random in each iteration. The algorithm will stop if no further improvement is found in certain iterations. The initial solution is obtained by setting each school's dismissal time equal to its earliest dismissal time and solve the scheduling problem using Kim et al.'s Type-I formulation [4]. Under the assumption that trips' start times are known (equal to the school dismissal time), the scheduling problem is formulated as a modified assignment problem, which can be solved using the Hungarian Algorithm that has an $O(n^3)$ time-complexity [5].

3 Result and conclusion

The S-TARS and the algorithm are used to solve the HCPSS problem. We tested three different bell time proposals: ES first (elementary schools start the first), MS first (middle schools start the first); HS first (high schools start the first). All the results are shown in Figure 1. The HS first yields the best solution. Under HS first assumption, all high schools start at 7:25 a.m., all middle schools start between 8:15 a.m. and 8:45 a.m. and all elementary schools start between 8:00 a.m. and 9:15 a.m. The number of buses is reduced from 324 to 295. This 29 bus saving corresponds to 8.9% improvement of the solution, and it saves approximately 2 million dollars annually for Howard County. The same method was applied to optimize the bus transportation system for Aurora Joint School District in Colorado with 39 schools and 416 trips. We can reduce 58 buses down to 37 buses, which is a significant 36% improvement.

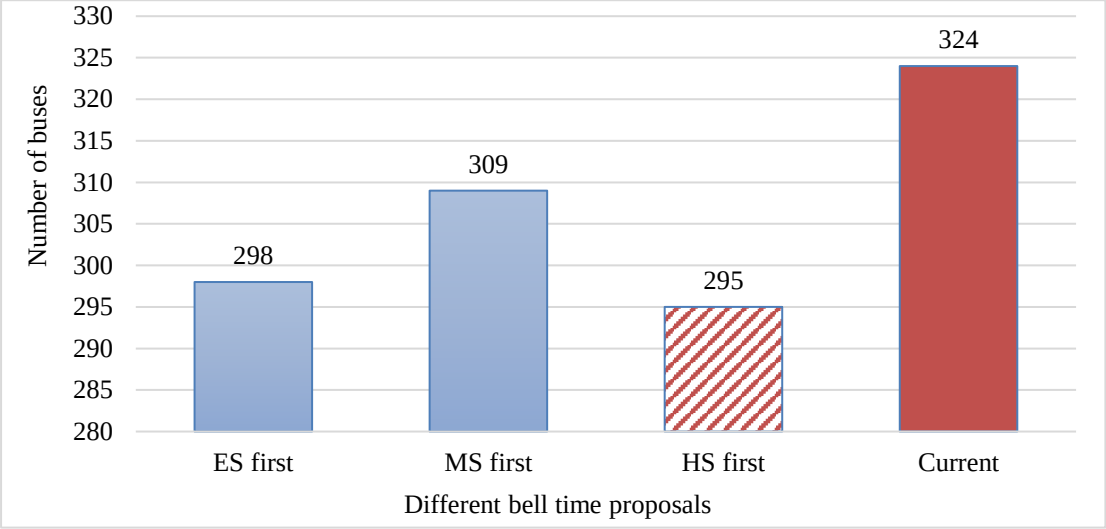


Figure 1 Solution comparison of HCPSS

This paper showed the huge benefit of optimizing the school bell time and route schedule as one joint problem. With realistic simplifications, the novel school-based formulation can significantly reduce 91% binary variable the and 98% constraint from the trip-based formulation. The local search-based algorithm is shown to be effective on large scale problem with stunning performance. The methodology has been successfully implemented in two real-world problems.

References

[1] Desrosiers, J., Ferland, J.A., Rousseau, J.-M., Lapalme, G., Chapleau, L., 1981. An overview of a school busing system. In: Jaiswal, N.K. (Ed.), Scientific Management of Transport Systems. North-Holland, Amsterdam, pp. 235–243.

[2] Fügenschuh, A., 2009. Solving a school bus scheduling problem with integer programming. European Journal of Operational Research, 193(3), pp.867-884.

[3] Fügenschuh, A., 2011. A set partitioning reformulation of a school bus scheduling problem. Journal of Scheduling, 14(4), pp.307-318.

[4] Kim, B.I., Kim, S., and Park, J., 2012. A school bus scheduling problem. European Journal of Operational Research, 218(2), pp.577-585.

[5] Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems, Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38, 1957 March.

A mathematical model and a solution algorithm for the electric vehicle routing problem with non-stationary battery swapping

Ramin Raeesi

Centre for Transport and Logistics (CENTRAL),

Department of Management Science, Lancaster University Management School,

Lancaster LA1 4YX, UK

Email: r.raeesi@lancaster.ac.uk

Konstantinos G. Zografos

Centre for Transport and Logistics (CENTRAL),

Department of Management Science, Lancaster University Management School,

Lancaster LA1 4YX, UK

1 Introduction

The ever-increasing contribution of Urban Freight Distribution (UFD) to urban traffic congestion and pollutants emissions has drawn attention to the use of Electric Commercial Vehicles (ECVs) that promise zero local emissions for city logistics. ECVs adoption for UFD, however, is still significantly constrained by their (i) high acquisition cost, (ii) reduced driving range, (iii) long recharging time, and (iv) scarce and unevenly scattered Charging Stations (CSs).

In order to address the primary concern with running on an ECV fleet which corresponds to "range anxiety", the existing literature on the Electric Vehicle Routing Problems (ECVRPs) has focused on the consideration of available CSs in the network, and thus introducing minimal vehicle detours in the ECV route to visit CSs if required [1-5]. While this can aid companies to plan their routes ahead and identify the need for recharging at potential CSs in advance, in the presence of realistic time windows the solutions yielded by ECVRPs might be either infeasible or too expensive in terms of the number of ECVs required and the total distance to travel. To address these shortcomings, in this paper, we turn our attention to new technological developments in the area of electric vehicles pertinent to swapping or recharging the ECV battery on-the-fly using a mobile Battery Swapping (recharging) Van (BSV). As described in [6], the development of a new fast battery-swapping device installed on a BSV opens up new possibilities to freight distribution with ECVs by providing an *"active"* battery-swapping mode. Therefore, in this study, we introduce and study the Synchronised Electric Vehicle Routing Problem with Non-Stationary Battery Swapping (SEVRP-NSBS), in which if a swap is required for an ECV to be able to carry out its route, a BSV is sent to visit the corresponding ECV at a designated point and time. It must be mentioned that in the proposed SEVRP-NSBS we retain battery recharging at CSs as the primary solution to routing a fleet of ECVs, and given the potential high acquisition cost of BSVs, we only propose to use them when it is not possible to satisfy customers time windows, or it is *optimal* to employ a BSV rather than visiting a CS. The contribution of this paper is multi-fold: (i) the SEVRP-

NSBS is introduced and formulated as a Mixed Integer Linear Programming (MILP) model, (ii) new analytical results, leading to a Graph Reduction Approach (GRA), are developed to identify a priori all eligible paths passing through one or several CSs between every pair of customers, (iii) a significantly strengthened alternative formulation of the problem is developed based on the proposed GRA that can solve some of the previously unsolved EVRPTW instances to optimality, and (iv) a two-stage memetic solution algorithm is developed to solve practical instances of the SEVRP-NSBS in a reasonable computational time.

In the remainder of the paper, we present a formal description of the problem and the model, a high-level exposition of the solution algorithm, and some preliminary results.

2 The SEVRP-NSBS

The SEVRP-NSBS is defined on a complete, directed graph $G = (N, A)$, where N is the set of network nodes and $A = \{(i,j)|i,j \in N, i \neq j\}$ is the set of directed arcs. The set $N = \{N_0 \cup N_1 \cup N_2\}$ is comprised of the depot $N_0 = \{0, n+m+1\}$, with $\{n+m+1\}$ being a dummy copy of $\{0\}$, customer nodes $N_1 = \{1,2,\dots,n\}$, and CSs $N_2 = \{n+1,\dots,n+m\}$. Each customer $i \in N_1$ is associated with a certain demand q_i to be delivered within its pre-determined hard time window, denoted by $w_i = [e_i, l_i]$, with service time s_i. The depot working hours, which is considered as the planning horizon, is denoted by $T = w_0 = [e_0, l_0]$. To each arc $(i,j) \in A$, a distance d_{ij}, and a travel time t_{ij} is attributed. There is a fleet of homogeneous ECVs, K_1 and a fleet of homogeneous BSVs, K_2 located in the central depot. The fleet of all vehicle types is denoted by $K = \{K_1 \cup K_2\}$. To each ECV $k \in K_1$ a maximum payload Q_1, a battery capacity B_1, a daily hiring fixed cost c_1, and an energy consumption rate per unit distance travelled r_1 is attributed. Each BSV $k \in K_2$, on the other hand, can carry a maximum number of batteries Q_2, has a daily hiring fixed cost c_2, a battery capacity B_2, and an energy consumption rate r_2. The time spent for recharging an ECV at a CS is dependent on the State of the Charge (SOC) of the battery upon arrival at the CS and inverse recharging rate is denoted by g. Battery swapping must be carried out at one of the network nodes, and realistically it cannot be done simultaneous with the ECV providing service at a customer. Hence, battery swapping can only start once ECV service is over. The arrival time of the BSV at the swapping location must be therefore synchronised with the ECV service finish time. However, the BSV can arrive earlier and wait till swapping starts. It is assumed that swapping takes p time units.

The aim of the SEVRP-NSBS is to determine an optimal composition of ECVs and BSVs in the fleet to operate routes that start and finish at the depot and serve every customer exactly once within their pre-defined time-windows, without violating vehicle capacities, battery level availability, and working day limits, such that the vehicle hiring cost and the total distance of the routes are minimised.

The MILP formulation of the problem works with three decision variables: the binary decision variable $x_{ij}^k \in \{0,1\}$ is equal to 1 iff vehicle $k \in K$ traverses arc $(i,j) \in A$; the continuous variable y_i^k denotes the service start time (customer service in case of ECVs and swapping service in case of BSVs) of vehicle $k \in K$ at node $i \in N$; and finally, the continuous variable u_i^k denotes the battery level of vehicle $k \in K$ upon its departure from node $i \in N$. Due to space limitation, we avoid a full presentation

of the MILP, and as most of the constraints remain similar to the standard constraints for the EVRPTW, we discuss briefly a couple of distinctive modelling features here:

- $y_i^k + (d_{ij} + s_i)x_{ij}^k + \boldsymbol{p}\sum_{k\in K_2}\sum_{j\in N} x_{ji}^k - (l_0 + \boldsymbol{p})(1 - x_{ij}^k) \le y_j^k,\ \forall k \in K_1, i \in N_1 \cup \{0\}, j \in N\backslash\{0\}$: These constraints determine the service start time at a customer by an ECV. Based on these constraints, the departure time from the upstream node is determined by the service start time plus the service time and swapping time if it is occurring at the customer.
- $y_i^k + (d_{ij} + \boldsymbol{p})x_{ij}^k - l_0(1 - x_{ij}^k) \le y_j^k, \quad \forall k \in K_2, i \in N_1 \cup \{0\}, j \in N_1 \cup \{n+m+1\}$: These constraints determine the battery swapping service start time at a customer by a BSV.
- $y_i^{k_1} + s_i - l_0(1 - \sum_{j\in N} x_{ji}^{k_2}) \le y_i^{k_2} \le l_0 \sum_{j\in N} x_{ij}^{k_2} + x_{ji}^{k_2}, \quad \forall k_1 \in K_1, k_2 \in K_2, i \in N_1$: These constraints synchronise the battery swapping service start time of a BSV with the service finish time of the corresponding ECV at the customer location.
- $u_j^k \le u_i^k - r_1 d_{ij} x_{ij}^k + B_1 \sum_{k\in K_2}\sum_{j\in N} x_{ji}^k + B_1(1 - x_{ij}^k), \quad \forall k \in K_1, i \in N_1, j \in N\backslash\{0\}$ and $\sum_{j\in N} r_1 d_{ij} x_{ij}^k \le u_i^k \le B_1 \sum_{j\in N} x_{ij}^k,\ \forall k \in K_1, i \in N_1 \cup \{0\}$: These constraints together tune the dependency of the battery level of an ECV on the distance travelled and any determined battery swapping.

3 The solution algorithm

A primary complication in addressing the EVRPTW with CSs is to determine which CS(s) should be selected, and where should the selected CS(s) be placed in the routes. In this study, we propose new analytical results leading to a GRA, based on which we can identify and discard all proven to be redundant paths passing through one or several CSs between a pair of customers, and hence only retain the remaining paths as eligible paths, and develop closed form formula for their attributes. As a result of the GRA, we are able to eliminate all CSs from the graph and work on a multi-graph of the eligible paths, and hence reduce the EVRPTWs with CSs to a VRPTW with alternative paths which can be more efficiently handled. The formulation of the SEVRP-NSBS and existing formulations for the EVRPTW with CSs can be significantly strengthened using the proposed GRA, and by just putting the formulation into the solver, it is possible to solve some of the EVRPTW instances that have been remained unsolved.

To solve SEVRP-NSBS instances of practical sizes in a reasonable computational time, we are proposing a two-stage Memetic Algorithm (MA) that solves the problem on the multi-graph resulted from the application of the GRA. While the proposed MA uses the common steps of initialisation, parent selection and crossover, education, intensification, and survivor selection, it introduces a new feature of the 'Routes Inventory' and an 'Inventory-to-Route' feature to restart the algorithm with high quality solutions and avoid the algorithm to get trapped in local optima. In the first stage of the proposed algorithm, the problem is optimised by only using the ECVs. If a dummy path that corresponds to a swap is present in the solution returned by the first stage of the algorithm, the second stage of the algorithm is informed that swaps are required and BSVs should be dispatched. The second stage problem, however, is a very small VRPTW that can be solved very quickly.

4 Preliminary results

In order to demonstrate very briefly the benefits of using non-stationary battery swaps, in this section we use 6 instances of size 25 that are adopted from Desaulniers et al. [7] and modified by multiplying

the g value by 3 to make them suitable for SEVRP-NSBS. The result of applying the GRA-based MILP for solving these instances is shown in Table 1. In this table, the heading EVRPTW implies approaching the problem as an EVRPTW with CSs, and the SEVRP-NSBS heading shows the effect of considering non-stationary battery swaps. In the case of the SEVRP-NSBS formulation, the number of BSVs and ECVs employed and the total distance they travel in the solution are reported separately and altogether as total. The column 'No. Swaps' shows the total number of battery swapping scheduled.

	EVRPTW		SEVRP-NSBS						
Instance	No. ECVs	Distance	No. BSVs	BSVs Distance	No. Swaps	No. ECVs	ECVs Distance	Total Vehicles	Total Distance
C101	8	780.15	1	53.47	3	7	625.38	**8**	**678.85**
C102	Infeasible		1	28.95	2	6	557.23	**7**	**586.18**
C105	8	628.02	1	65.01	2	6	530.84	**7**	**595.85**
C106	8	778.07	1	45.71	3	6	597.63	**7**	**643.35**
C107	7	574.59	1	26	1	6	525.94	**7**	**551.94**
C108	6	566.47	1	53.37	2	5	510.82	**6**	**564.2**

As it can be seen in the table, in all the 6 instances considered, scheduling battery swaps by BSVs instead of visiting CSs costs less in terms of both the total number of vehicles needed and the total distance travelled. Moreover, in the case of the second instance, i.e. C102, it is not even possible to find a feasible solution to the problem by only visiting CSs.

More extensive experimentation results on the model and the algorithm will be presented in our presentation.

References

[1] M. Schneider, A. Stenger, and D. Goeke "The electric vehicle-routing problem with time windows and recharging stations", *Transportation Science* 48(4), 500-520 (2014).

[2] D.J. Goeke and M. Schneider "Routing a mixed fleet of electric and conventional vehicles", *European Journal of Operational Research* 245(1), 81-99 (2015).

[3] M. Bruglieri, F. Pezzella, O. Pisacane, and S. Suraci, "A variable neighborhood search branching for the electric vehicle routing problem with time windows", *Electronic Notes in Discrete Mathematics* 47, 221-228 (2015).

[4] M. Keskin and B. Catay "Partial recharge strategies for the electric vehicle routing problem with time windows", *Transportation Research Part C: Emerging Technologies* 65, 111-127 (2016).

[5] A. Montoya, C. Guret, J.E. Mendoza, and J.G. Villegas "The electric vehicle routing problem with nonlinear charging function", *Transportation Research Part B: Methodological* 103, 87-110 (2017).

[6] S. Shao, S. Guo, and X. Qiu "A Mobile Battery Swapping Service for Electric Vehicles Based on a Battery Swapping Van", *Energies* 10(10), 1667 (2017)

[7] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider "Exact algorithms for electric vehicle-routing problems with time windows" *Operations Research* 64(6), 1388-1405 (2016).

Forecasting a freight carrier's demand for container shipments

Greta Laage (corresponding author)
CIRRELT and Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal, Canada
Email: greta.laage@polymtl.ca
Emma Frejinger
CIRRELT and Department of Computer Science and Operations Research
Université de Montréal, Canada
Gilles Savard
IVADO and Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal, Canada

1 Introduction

Forecasting demand for container shipments over time for different origin-destination pairs (OD) is a problem of high significance to many transport applications. It is challenging for several reasons, for example, demand for different OD pairs depend on each other (i.e., spatial correlation), demand variations are linked to global supply chains and economic factors, the number of OD pairs is potentially large and there are several types of containers. We take the perspective of a freight carrier having access to historical data of past shipments in its network. More precisely, we train deep learning algorithms – multilayer perceptron (MLP) and recurrent neural network (RNN) – on the historical data with the objective of accurately forecasting future daily demand. Our carrier of interest is the Canadian National Railway Company (CN), one of the largest rail carriers in North America. The forecasts will be used as inputs for block planning of intermodal traffic [1]. The latter is a tactical problem defined over a weekly planning horizon that in an operational setting is updated daily. We therefore predict the daily quantity of containers of each type to be carried on each OD pair over the next seven days for short-term forecasts, or several weeks for the medium-term ones.

Demand forecasting has been the focus of many studies that take a perspective different from

ours. Namely, the aim is to forecast the freight flows in an entire transport system, hence incorporating many actors and stakeholders [2]. In this context the challenge lies in obtaining accurate data on production-consumption matrices and distributing the freight flows on different transport modes and on different routes in the network. The challenges we face are different since we take the perspective of a single carrier. Our problem has some similarities with the prediction of demand in the passenger airline industry [3], however, with the additional challenge that there are several types of containers that in turn require different equipment (here railcars) and loading constraints. It is a specific application of multivariate time series analysis. Thanks to the availability of large sources of data, forecasting methods have been shifting from classic statistical models to machine learning approaches. In this context deep learning is particularly promising since it can approximate any high-dimensional complex function [4].

The literature on machine learning approaches for our problem is scarce. Closest to our work is the deep learning models developed for short-term traffic flow prediction which is also a multivariate time series including both time and space correlation. Two architectures, the convolutional and the recurrent long short-term memory (LSTM), have been combined to model respectively spatial and temporal features of traffic flow on a freeway corridor [5]. However, this work estimates the future traffic between each location point of one highway at the next 5 minutes time point. This is still quite different from our network perspective and multi-step forecasting horizon.

In summary, we are unaware of any work taking a single carrier perspective and using deep learning models to forecast demand for container shipments. We contribute to the literature by devising such models to produce short-term and medium-term forecasts. Furthermore, we report results based on a real case study from one of the largest railroads in North America. We provide in Section 2 a description of the prediction problem and the available data. We introduce our deep architectures and computational results in Section 3.

2 Prediction Problem and Data

Our multivariate time series prediction problem consists in forecasting the number of containers of each type to carry daily on each OD pair of the network using historical data. It requires taking into account spatial and temporal correlation among other challenges described below.

Demand varies over time and long-term dependencies as well as periodicity can be OD specific. As operations are defined weekly, we need to model OD specific temporal correlation within a week and among weeks. Temporal variations of demand often depend on variations from other OD pairs. For instance, a peak of demand for an OD pair might lead to a higher traffic later on the DO pair to bring back empty containers. This results in the significant presence of correlation in rail transportation networks.

Our network planning problem is large scale. The application gathers some 30 of intermodal

terminals and multiple types of containers. Thus, daily forecasts for a week-long horizon of each type of container on each OD pair form a large 3D matrix. Such complex data contain non-linearity that is difficult to model with classic, often linear, time-series models.

We hold millions of data records describing the container shipments over the past four years on our industrial partner's network. However, observed demand does not always match real demand and we talk about censored data. Censoring can result from both high-level competitive markets and carrier's operations. Containers left at origin due to a lack of resources delay demand to a later day. In this case, observed demand for the first day underestimates real demand but overestimates demand for the later day. Transportation networks are typically unbalanced (high versus low demand OD pairs) which allows us to identify uncensored and censored OD pairs in our application. We can therefore categorize the OD pairs and devise a separate forecasting model for those that are uncensored. In this work we focus on these *uncensored OD pairs*.

Let C denote the set of types of containers, E the set of OD pairs and $y^t_{e,c}$ the number of containers of type c on $e \in E$ at time t. In this work, we define forecasting models which output, at each day j, an estimate of $\{y^{j+1}_{e,c}, .., y^{j+7}_{e,c}, c \in C, e \in E\}$. Those models take as inputs previous data from day j including demand on multiple OD pairs and economic indicators. We identified the latter after discussions with our industrial partner. We propose using deep neural networks as they are able to model non-linear and high-dimensional problems. Our objective is to get the most accurate forecasts, we hence compare the performance of several different models that we describe in the following section.

3 Learning architecture and computational results

We define two deep learning architectures to forecast intermodal freight demand which we adjust to model temporal and spatial correlation: the MLP and the recurrent LSTM. To our knowledge, each designed architecture is a new approach for the forecasting problem at hand. We describe in this section each architecture and experimental results.

To model temporal features, inputs for the first architecture include previous observations which number is a hyperparameter to be defined. While this model is faster to train than the recurrent one, it does not contain a memory. One of the most successful architectures to characterize long-term dependencies is the LSTM which learns both short-term and long-term memory.

To model spatial correlation, we compare two designs for both architectures described above. In the first one, one neural network is trained per OD pair and inputs include historical data from the target and other strategically selected OD pairs. In the second one, the carrier's network is separated into subsets of OD pairs and one neural network is trained per subset. Outputs are forecasts for each OD pair in the subset and inputs include historical data from those OD pairs.

To compare performances of the different architectures we decompose the data set, as standard,

into training, validation and test sets. We use either the mean-squared error (MSE) or the mean absolute error (MAE) to train neural networks. We assess the performance of the architectures using the mean absolute percentage error (MAPE) as the total quantities of each type of container are different and the same MAE for two types of containers could have different interpretations. We select the set of hyperparameters which performs best on the validation set. We have extensive numerical results for the different architectures, OD pairs, types of containers and forecasting horizon. In brief, the results show that the architecture giving the best performances depends on the OD pair and type of container to predict. On the validation set, the lowest MAPE achieved so far is 15.22% for the MLP and 15.52% for the LSTM for the 7 days ahead forecasting horizon.

At the conference, we will present a comparison between the different deep learning models as well as a comparison with classic benchmark time series models such as the autoregressive model.

4 Conclusion

We trained different deep learning algorithms addressing the problem of intermodal demand forecasting for a carrier and its numerous challenges. Those include in particular the interdependence of OD pairs, the temporal variations of demand and the presence of several types of containers. This work is part of an ongoing research effort where the end goal is to link the demand predictions with an intermodal block planning model.

We gratefully acknowledge the close collaboration with CN and the funding through the CN Chair on Optimization of Railway Operations.

References

[1] G. Morganti, S. Bisaillon, T.G. Crainic and E. Frejinger, "Block and car planning for intermodal rail", 7th ODYSSEUS Conference, Italy, June 2018.

[2] G. De Jong, H. Gunn and W. Walker, "National and international freight transport models: an overview and ideas for future development", *Transport Reviews*, 24(1),103-124, 2004.

[3] L. Weatherford, "The history of forecasting models in revenue management", *Journal of Pricing and Revenue Management*, 15(3-4), 212-221, 2016.

[4] M.G. Karlaftis and E.I. Vlahogianni, "Statistics versus neural networks in transportation research: differences, similarities and some insights", *Transportation Research Part C: Emerging Technologies*, 19(3), 387-399, 2011.

[5] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework", *arXiv preprint arXiv:1612.01022*, 2016.

Optimizing the Training Transfer of Junior Soccer Players

Christian Jost*, Alexander Döge†, Sebastian Schiffels*, Rainer Kolisch*

*TUM School of Management, Technical University of Munich

†CoE Excellence & Innovations BASF SE

Email: christian.jost@tum.de

1 Introduction

Soccer fascinates millions of people around the globe. Its immense popularity is one of the driving factors for the high economic relevance of national leagues, like the German Bundesliga, increasing its annual turnover to 3.89 billion USD in 2018. In this multi billion dollar business, rising player transfer cost make the promotion of young talents a key factor for the soccer clubs' long term success. The youth academy of the Bundesliga club TSG 1899 Hoffenheim (TSG) is one of the most renown soccer training academies in Germany, with players from all over the country's southwest attending the training. As no sufficient public transport to the training centers exists, the TSG provides a bus transfer service for its U12-U19 players. Manual scheduling of several buses is a complex and time-consuming task, which currently leaves many players unserved. Many unserved players resort to private transport like parental car pooling, while some players choose to reside permanently at the TSG's boarding school. Our approach improves this situation by increasing the number of players using the bus service, reducing their parents' time expense and allowing their children to live with their families while pursuing their dreams. Furthermore, in the highly contested market for youth players, facilitating easy training access yields a significant competitive advantage for the TSG. On an operational level, our approach reduces the weekly planning effort from several days to a few minutes, allowing the TSG's staff to quickly adapt the routing to changes in pickup demand.

The proposed framework handles the multi period transfer problem including driver-player assignment consistencies throughout the season as well as the single day transfer problem. For the single day transfer problem we extend the team orienteering problem (TOP) [1] to optimize the bus routing on a single training day. We consider ten buses and over 100 players in 70 locations, some of which live more than 100 km ($\sim$ 62 miles) away from the training center. Given the limited

seating capacities of four to eight passengers per bus, and a maximum ride duration of two hours per player, players are prioritized according to their age group: The older the player, the more advanced his career, the higher the priority for transport. The tours are calculated maximizing the sum of the priorities of the picked up players.

In the multi period transfer problem, we extend the single day formulation to the entire 235 training days of the season. While maximizing the sum of the priorities of the picked up players, we aim to keep the driver-player assignment consistent across training days for several reasons: (1) A personal driver-player relationship and mutual trust are important safety factors, especially for children underage; (2) players should know their designated driver and vice versa in order to communicate changes in pickup time or place on short notice, and (3) interviews with the drivers show that transporting the same players increases their work satisfaction, which is especially important as some of them work on a voluntary basis. For the TSG, assignment consistency is a necessary requirement. Without it, tours and therefore driver to player assignments vary significantly due to the differences in training schedules and pickup requests across training days.

The player transport problem combines the TOP with the consistent vehicle routing problem (ConVRP) [2]. The team orienteering problem as a variant of vehicle routing with profits, has been applied successfully to a wide range of applications, finding the profit maximizing tours given a limit in capacity or travel time [3]. The ConVRP literature can be divided into integrated approaches, where consistency is part of the objective function, and multi stage approaches, where a tour template is derived and used as a blueprint for constructing similar tours. Our approach has a multi stage structure, with a bus schedule derived at the beginning of the season, and daily schedules derived throughout the season, as updated pickup demands become available.

To the best of our knowledge, the proposed player transfer approach is the first to look at the interplay between assignment consistency and the profit maximization objective of the TOP. The pickup choice is depending on the profit (priority) increase, given a certain resource consumption, as well as the impact of the pickup on the overall assignment consistency. The proposed approach provides a tool for analyzing this trade-off and its effects on a real-world optimization problem.

Section 2.1 presents the single day transfer problem and the Tabu Search procedure. In Section 2.2 we solve the multi period transfer problem by deriving a template and resolving it for each daily routing problem of the season. Section 3 concludes by discussing the preliminary results.

2 The Player Transfer Problem

2.1 Single Day Transfer Problem

The single day transfer problem is an extension of the general TOP presented in [1]. The solution space is represented by a fully connected, directed graph $\mathcal{G}$ where each node $i \in \mathcal{V}\backslash\{0\}$ corresponds

to a player requesting transfer. All tours start and end at the training center $i = 0$. The number of tours is limited to the number of vehicles. Each vehicle has a maximum seating capacity and a maximum tour duration applies. The objective is to maximize the sum of the priorities of the picked up players given the capacity restrictions.

Small instances of a mixed integer problem formulation can be solved to optimality using CPLEX. For real-world instances, we implemented a Tabu Search (TS), favored by its frequent use in TOP literature, and its applicability at the TSG. Starting from an initial solution, the TS improves the daily transfer solution by changing the pickup sequences and interchanging players between tours. To comply with the capacity restrictions, players can be removed from a tour and put on a candidate list, or taken from the candidate list and inserted into a tour, given excess capacities. The solution is represented by a single tour for each vehicle, indicating the players to be picked up and the pickup order.

2.2 Multi Period Transfer Problem

With our multi period approach we yield a solution to the single day transfer problem for each training day, while providing the desired level of tour consistency throughout the season. Since we are interested in assigning each player $i \in \mathcal{V}$ to the same driver $k \in \mathcal{K}$ on every day $t \in \mathcal{T}$, we measure assignment consistency as follows: $\left(\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{V}} y_{t,i,k_i^*}\right) / \left(\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{V}}\sum_{k\in\mathcal{K}} y_{t,i,k}\right)$, where $y_{t,i,k} \in \{0,1\}$ indicates the pickup decision and k_i^* is the driver with the most pickups of player i. In case every player $i \in \mathcal{V}$ is picked up only by driver k_i^*, the consistency measure yields 100 %. However, given the limited resources, assignment consistency is in conflict with the priority maximization objective. Therefore we aim for a consistency level high enough to satisfy the TSG's requirements, while at the same time allowing for sufficient flexibility to achieve a high priority served.

For our approach, we use the pickup request forecast provided by the TSG at the beginning of the season and select a subset $\mathcal{M} \subseteq \mathcal{V}$ of players with the number of pickup request greater than a threshold n. We apply greedy construction in combination with the aforementioned TS to build a template of tours containing only players $i \in \mathcal{M}$. This template provides a structure, which we use to construct the daily tours during the season.

For each day $t \in \mathcal{T}$ of the season, we resolve the template by first removing all the players $i \in \mathcal{M}$ which do not request a transfer on day t. The pickup decision for the remaining players in the template is fixed ($y_{t,i,k} = 1$). Second, using cheapest insertion, we insert the players $i \notin \mathcal{M}$ that request a transfer on day t. This provides us with an initial solution. Finally, we use TS as described in section 2.1 to obtain the daily tours. Neighborhood moves switching players $i \in \mathcal{M}$ to another tour or removing them from their tour are prohibited. Therefore, players $i \in \mathcal{M}$ are always part of the same tour on each day they request a training transfer. This process is conducted for every day of the season, whenever new information on the pickup requests becomes available.

When constructing the daily tours, players in set $\mathcal{M}$ act as corner stones, forcing the tours to evolve into certain geographical regions every day of the season. This creates assignment consistency, while at the same time providing pickup flexibility when assigning players not part of $\mathcal{M}$. By systematically varying the pickup request threshold n, we increase/decrease the number of players in $\mathcal{M}$. In general, a higher consistency level can be observed for a lower threshold n and a larger subset $\mathcal{M}$. Therefore, through variation of n, we derive a pareto front showing the trade of between consistency and priority served.

3 Results

Preliminary result for the season 2018/2019 data show that for $\mathcal{M} = \emptyset$ we yield a base value of 49% assignment consistency. The stepwise increase of $\mathcal{M}$ up to 31% of all players, results in a maximum consistency level of 87%. Further increasing the size of $\mathcal{M}$ leads to a conflict between resource constraints and the fixed assignment of players $i \in \mathcal{M}$ during daily routing.

The increase in assignment consistency comes with a decrease in priority served. However, even at high consistency levels our approach provides sufficient flexibility to mitigate this effect. When increasing assignment consistency from 49% to 87%, the loss in priority served is only 2%. In terms of unversed players, this is the equivalent of one unserved medium priority player per day.

For the training transfer case we show, that high assignment consistency can be achieved at low priority losses. At the TSG, our approach replaced the manual solution and increased the priority served by up to 26%.

References

[1] C. Archetti, M.G. Speranza, D. Vigo "Vehicle Routing Problems with Profits", in *Vehicle routing: Problems, Methods, and Applications*, P. Toth and D. Vigo (eds), 223-248, SIAM, Philadelphia, USA, 2014.

[2] A.A. Kovacs, S.N. Parragh, R.F. Hartl, "A template–based adaptive large neighborhood search for the consistent vehicle routing problem", *Networks* 63, 60–81 (2014).

[3] C. Groër, B. Golden, E. Wasil, "The consistent vehicle routing problem", *Manufacturing & Service Operations Management* 11, 630–643 (2009).

Combinatorial Auction with Bidder-Defined Items for Fractional Ownership of Autonomous Vehicles

Mahdi Takalloo **Aigerim Bogyrbayeva** **Hadi Charkhgard**
Changhyun Kwon
Department of Industrial and Management Science Engineering
University of South Florida
Email: chkwon@usf.adu

1 Introduction

When autonomous vehicles (AVs) are introduced to the consumer markets, fractional ownership is expected to be a form of owning a car. Currently, co-owning a conventional vehicle with friends is not easy, although it can certainly reduce the cost of owning a car. For example, if I want to use a car for commuting and my friend wants to use it while I am at work, my friend must come to my workplace to pick the car up, which requires another form of mobility. With AVs, the co-owned car can travel autonomously from my work to my friend's location. Therefore, we envision that AVs will be co-owned widely and new markets will be created accordingly.

The main question we address in this paper is: How can we design a marketplace that connects customers and enables fractional ownership? In particular, we design a *Combinatorial Auction* (CA) for fractional AV ownership. CAs are suitable mechanisms to sell items in packages, instead of single items. In transportation, CAs have gained attention for selling airport departure and arrival slots [3], assigning trucking carriers [1], assigning city bus routes [1], and selling tradable permits in ride-sharing market [2]. We propose a new application of CAs: the fractional AV ownership market.

In the proposed market, the auctioneer is a car manufacturer or leasing company who sells AVs, and the bidders are customers who co-lease a car. In the proposed market, first, bidders submit their time-slot packages. Next, the auctioneer pools all the bids and solves the *Winner Determination Problem* (WDP) to determine the winners. The winners are awarded the right to use the same vehicle in these time-slots within a week for a certain period.

We design a combinatorial auction for fractional ownership of autonomous vehicles, which is of a novel type with continuous-time bidder-defined items. On the contrary to the most existing CAs where products are pre-defined discrete items, in the proposed auction, items are *bidder-defined*

and *continuous* time intervals. We show that the social welfare from the discrete-time approach does not monotonically increase with the number of discrete time slots, which makes discrete-time approximation difficult. We formulate the WDP, with numerically efficient reformulations, and develop an algorithm based on conflict graph and maximal cliques. Using the California Household Travel Survey, we verify the performance of the algorithm. When the WDP is solved sub-optimally, under the VCG mechanism, we show that the revenue of the auctioneer approaches to the optimally solved case as the optimality gap decreases, although the payment of each winner does not.

2 The Winner Determination Problem

Suppose $\mathcal{I}$ denote the set of bidders, and $\mathcal{V}$ denote the set of vehicles in the fractional AV ownership CA. Each bidder $i \in \mathcal{I}$ submits a set of bids $\mathcal{B}_i$. Each bid j includes the bidding price c_j, the set of trips $\mathcal{T}_j$, and the location of the bidder at the origin and the destination of each trip. Each trip $n \in \mathcal{T}_j$ is represented by the pair (s_n, e_n), where s_n is the start time and e_n is the end time of that trip. A parameter $r_{ike_ms_n}$ represents the time it takes for an AV to drive from bidder's i location at time e_m to bidder's k location at time s_n. We can formulate the WDP as follows:

$$\textsf{(P1)} \quad \max_{x_{jv}} \quad \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{B}_i} c_j x_{jv} \tag{1}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{B}_i}\sum_{v\in\mathcal{V}} x_{jv} \le 1 \qquad \forall i \in \mathcal{I} \tag{2}$$

$$e_m x_{jv} \le s_n + M(1 - x_{lv}) \qquad \forall i,k \in \mathcal{I}, j \in \mathcal{B}_i, l \in \mathcal{B}_k, \; m \in \mathcal{T}_j, n \in \mathcal{T}_l : s_m \le s_n \le e_m \tag{3}$$

$$e_m + r_{ike_ms_n} x_{jv} \le s_n + M(1 - x_{lv}) \qquad \forall i,k \in \mathcal{I}, j \in \mathcal{B}_i, l \in \mathcal{B}_k, \; m \in \mathcal{T}_j, n \in \mathcal{T}_l : s_n \ge e_m \tag{4}$$

$$x_{jv} \in \{0,1\} \qquad \forall i \in \mathcal{I}, j \in \mathcal{B}_i, \forall v \in \mathcal{V} \tag{5}$$

The decision variable x_{jv} is 1 if bid j is assigned to vehicle v and is 0 otherwise. Constraint (2) states that at most one bid from each bidder can be determined as a winner. Constraints (3) and (4) ensures that conflicting bids do not get matched. Since the building and solving (P1) is time consuming, we propose a conflict-based formulation, which can be constructed and solved faster.

3 Computational Method and Experiments

We can replace (3)–(4) in (P1) with the following conflict constraints:

$$x_{jv} + x_{lv} \le 1 \quad \forall v \in \mathcal{V}, i,k \in \mathcal{I}, j \in \mathcal{B}_i, l \in \mathcal{B}_k : j,l \text{ are conflicting} \tag{6}$$

and we call the new formulation (P2). Constraint (6) requires finding all conflicting bids that have overlapping trips. We can design an efficient algorithm to find the set of conflicting bids in polynomial time. Once we determine the conflicting bids, we can solve the WDP by CPLEX or any other integer programming solver.

To find a high-quality solution for large-sized instances in a short time, we develop a greedy algorithm in which we decompose the CA problem to a $|\mathcal{V}|$-round single vehicle CA. At each round, considering the set of remaining bidders, we solve the WDP for a single vehicle and find the winners. Then, we update the set of bidders by excluding the winners from the list of bidders and go to the next round. This procedure continues until we assign all the vehicles to the bidders.

To show the quality of the solution obtained from the greedy algorithm, we propose a good relaxation of (P2), which is based on maximal-cliques. Constraint (6) can be viewed as a simple *clique* constraint. We can derive a better formulation by replacing Constraint (6) with stronger constraints based on maximal cliques. We introduce the following relaxation of problem (P2):

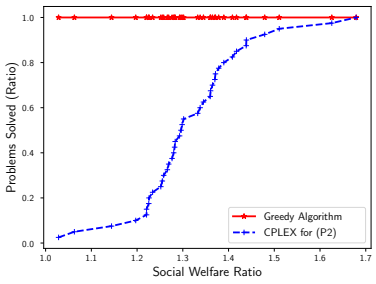
$$
\begin{aligned}
(\mathsf{R}) \quad \max_{y_j} \quad & \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{B}_i} c_j y_j && (7)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{B}_i} y_j \le 1 \quad \forall i \in \mathcal{I} && (8)\\
& \sum_{j\in\mathcal{C}_m} y_j \le |\mathcal{V}| \quad \forall m \in \mathcal{M} && (9)\\
& y_j \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{B}_i && (10)
\end{aligned}
$$

where $\mathcal{C}_m$ is the set of all maximal cliques. We solve problem (R) to compute a dual bound for (P2). Since enumerating all maximal cliques is computationally expensive, we consider a subset.

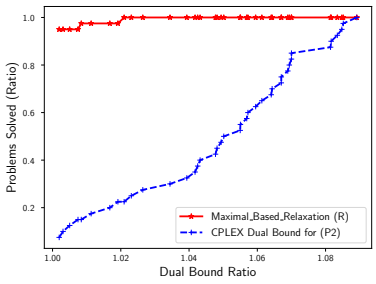
For numerical experiments, we use 2010–2012 California Household Travel Survey, which includes the travel information of 2908 vehicles in a week. We use this dataset to extract the trip schedules of bidders and to generate the instances. We compare the performance of the proposed greedy algorithm and the relaxation (R) with CPLEX. As the performance profile in Figure 1a shows, the greedy algorithm outperforms CPLEX for all instances. As Figure 1b represents, the dual bound found by (R) is smaller than the CPLEX dual bound for 37 out of 40 instances.

4 VCG Payments and Suboptimal Solutions

After winners are determined, we calculate the payments. We adopt the well-known VCG payments for the proposed auction. The VCG mechanism assumes optimal solutions of the WDP. When the WDP is solved sub-optimally, the desired properties of auctions such as incentive compatibility and rationality do not necessarily hold. We examine the impacts of suboptimal solutions in this paper. Moreover, we obtain the following bounds:



(a) Primal Solution Ratio



(b) Dual Bound Ratio (One hour time-limit)

Figure 1: Comparing primal solution and dual solution with CPLEX solution

Proposition 4.1 *Let p_i^* and p_i^ε denote the payments of a bidder i under the optimality and a optimality gap of ε, and R^* and R^ε denote the auctioneer's revenue under the optimality and an optimality gap of ε under the VCG mechanism, respectively. Then the following bounds hold:*

$$|p_i^\varepsilon - p_i^*| \leq \varepsilon \max\left\{Z^\varepsilon_{\mathrm{WDP}}, Z^\varepsilon_{\mathrm{WDP}_{-i}}\right\} + \max_{j\in\mathcal{B}_i} c_j, \tag{11}$$

$$|R^\varepsilon - R^*| \leq \varepsilon \max\left\{(|\mathcal{I}|-1)Z^\varepsilon_{\mathrm{WDP}}, \sum_{i\in\mathcal{I}} Z^\varepsilon_{\mathrm{WDP}_{-i}}\right\}. \tag{12}$$

This proposition implies that the revenue of the auctioneer will be close to the optimal case, although each winner's payment may not be. Through numerical experiments, we provide further insights on the payments and the revenue in this paper.

References

[1] Cantillon, E. and Pesendorfer, M. (2006). Auctioning bus routes: The London experience. pages 573–592.

[2] Hara, Y. and Hato, E. (2017). A car sharing auction with temporal-spatial od connection conditions. *Transportation Research Part B: Methodological*, pages 1–17.

[3] Rassenti, S. J., Smith, V. L., and Bulfin, R. L. (1982). A Combinatorial Auction Mechanism for Airport Time Slot Allocation. *The Bell Journal of Economics*, 13(2):402–417.

Improving Drayage Operations through a Realistic Optimization Model

Mahboobeh Moghaddam (Corresponding Author)

Australian Institute of Business and Economics

The University of Queensland, PACE Building, Woolloongabba QLD 4102

Email: m.moghaddam@uq.edu.au

Robin H. Pearce

School of Mathematics and Physics, the University of Queensland

Hamid Mokhtar

School of Information Technology and Electrical Engineering, the University of Queensland

Carlo Prato

School of Civil Engineering, the University of Queensland

1 Introduction

Inland drayage operations account for 20 to 80 percent of the total transportation cost of a shipping container, despite being the shortest distance segment of a container's trip [1], [2]. The considerable share of the drayage operations in the container transportation costs has placed the efficient planning of these operations at the centre of attention for shipping lines and transportation companies [3]. Moreover, the truck movements are often blamed for slowing the delivery, as well as increasing road congestion, environmental pollution, and road safety risks in the service area [4], [5].

These challenges have led to an increasing attention from the research community to seek solutions through optimization models for drayage operations. The objective of the models is to develop an optimal plan for a fleet of trucks to move a fixed set of containers during a specific planning horizon. However, due to the complexity of the drayage operations, most of the existing models include some simplifying assumptions. Examples include one truck-one container [6]–[8], homogeneous truck fleets [9], [10], one container size [11], a central point for all delivery and pickup [6], [12], uninterruptable pickup-and-delivery operation [7], [13], and considering only the live loading/unloading of containers where trucks have to stay with the container [10], [14].

The main implication of such assumptions is that the resulting formulation of the drayage operations cannot be applied to real world settings. To address this problem, we have developed a novel formulation of drayage operations by relaxing the above mentioned set of assumptions. The proposed model supports a heterogeneous truck fleet, multiple container sizes, live as well as separation mode un/load operations, and it has no restrictions on the number, location, or type of the stakeholders who need container transportation. Finally, the proposed model allows for combining multiple transport requests into one trip, which leads to the reduction of costs, emissions and congestion in the service area.

2 Methodology

We have formulated drayage operations as a mixed integer linear programming (MILP) problem. In the proposed model, the planning and optimization of drayage operations is performed from the point of

view of a transport company. The company owns a fleet of trucks with different capacities to carry the two common sizes of shipping containers, 20 and 40 feet, full or empty.

Transport requests are received from consignees and shippers with already defined origins and destinations. The requests can be divided into three types: single, live composite, and drop-in (separation mode) composite. A single request includes the origin and the destination of where to pick up and drop off a container, the associated time windows to visit the pickup and delivery locations, and the container identification number and size. A live composite request includes the container pickup location, the customer location to be visited after pickup, and the final delivery location of the container. Also, the time windows to visit the pickup, customer, and final delivery locations are defined as well as the service time required for the live (un)loading of the container at the customer site. A drop-in composite request (separation mode) has a structure similar to a live request; however, as the two parts of the request (pickup to customer, and customer to final delivery) do not need to be performed without interruption, a composite request is translated into two single requests for the planning system. Time dependency implies that the second single request should be executed after the first request.

The planning is performed for one planning horizon. The input to the planning is the set of all requests, and the output is a plan for each truck and its driver where and when to pick up and deliver a set of containers. The objective of the planning is to minimize the travel time for as many requests as possible, rather than for all the requests. Any infeasible request that cannot be fitted in the optimal plan, either has to be postponed if the timing allows or should be assigned to a contractor.

3 Data

The performance of the proposed mechanism is evaluated through simulation-based experiments, seeded from a real data set collected from a medium-size transport company, operating in the service area of the Port of Brisbane, Australia.

Regarding the sample size, three problem groups are considered: small, medium and large (as defined in Table 1). To design the fleet, we consider the ratio of the number of trucks to the number of requests. Based on the literature and similar experiments, the ratio is fixed at 0.27 and 0.48. The first ratio creates a tight capacity, and the second one provides a more relaxed capacity to accommodate the transport requests. To design a heterogeneous fleet, we consider the fleet to include trucks with capacities from 20 to 80 feet, equivalent to 1 to 4 TEUs (Twenty foot Equivalent Unit). To design the combination of trucks in a fleet, three types of fleets are designed: a Low Capacity (LC) fleet with mostly 20-ft and 40-ft trucks, a High Capacity (HC) fleet with mostly 60-ft and 80-ft trucks, and a Balanced Capacity fleet (BC) with a mixed balance of 20-ft, 40-ft, 60-ft, and 80-ft trucks. As a baseline for comparison, a homogenous fleet of 40-ft trucks is added to support the 20-ft and 40-ft containers. We have 24 problem sets in total, as presented in Table 2. For each problem set, 10 instances are generated. The time window to visit a pickup or drop off location is fixed at 30 minutes.

Table 1. Definition of sample size

Sample size	#requests	#customers	#depots
Small	37	15	1
Medium	56	20	2
Large	75	30	4

Table 2. Experiment Design Parameters

Parameter	Values
Sample size	Small, Medium, Large
Fleet type	LC, BC, HC, Baseline
Truck to request ratio	0.27, 0.48

4 Results

The results show that the success rate (SR: ratio of the number of requests satisfied to the total number of requests) is quite high across the three sample sizes; it is 92% in small, 87% in medium and 66% in large samples. As expected, the complexity of the large sample size makes it more difficult for the mechanism to match transport requests to the fleet.

Looking at the fleet type, SR is plotted in Figure 1. As we can see, in the small sample, the HC fleet has the highest SR (96%), closely followed by the balanced (95%) and the baseline (90%). The LC fleet has the lowest SR (86%). The medium sample shows similar results, except for the HC fleet where SR decreases to 73%. In the large sample, the SR of the high capacity fleet drops to 28%. The balanced fleet follows a less dramatic fall, being equal to 55%.

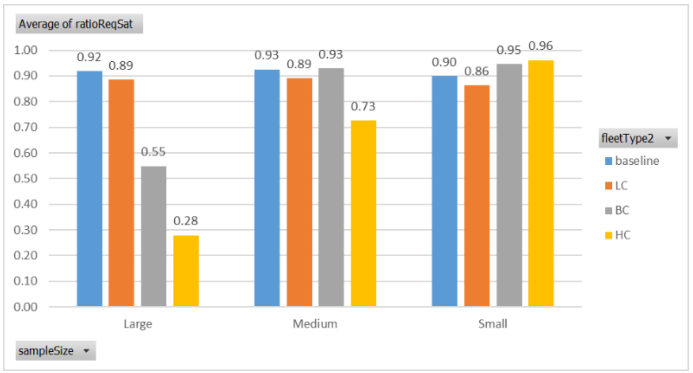


Figure 1. Average Success Rate (SR) versus sample size and fleet type

The decrease in SR can have two interpretations: (i) the inability of the HC fleet to accommodate transport requests; (ii) the complexity of the optimization problem prohibiting the high capacity fleet from achieving high SR. As with small sample size, the HC fleet is performing as well as other types of fleets, so it is more likely that the low SR in large and medium sample sizes is caused by the complexity of finding an optimal solution for drayage operations.

Checking the optimality gap (the difference between the best solution found by the end of the solve time and the objective bound, divided by the objective bound) in each sample size / fleet type group confirms our intuition about the complexity of the problem. As presented in Table 3 below, we can see that the average gap for the HC fleet-large sample is 3.71 (371%) indicating that the feasible solution in this group is on average far from optimality.

Table 3. Average gap of the feasible solution based on sample size and fleet type

Fleet Type / Sample Size	Baseline	LC	BC	HC	Grand Total
Large	0.01	0.01	1.17	3.71	1.19
Medium	0.00	0.00	0.06	0.53	0.15
Small	0.00	0.00	0.01	0.02	0.01
Grand Total	0.00	0.00	0.41	1.38	0.45

Considering the fleet utilization (the ratio of the number of trucks used in allocation to the total number of trucks available), the small sample has the highest fleet utilization (86%). The medium sample follows closely at 84%, and the large sample size has a fleet utilization rate of 71%.

The average fleet utilization rate is plotted against sample size and fleet type in Figure 2. In each sample size, the low capacity fleet has the highest fleet utilization, closely followed by the baseline. In small and

medium sample sizes, the balanced and HC fleet have close fleet utilization to other fleet categories. However, in large sample size, the high capacity fleet is behind by a relatively large gap (40% for the HC fleet compared to 93% for the LC fleet). The low utilization rate is again due to the complexity of the problem and the inability of the model to get close to optimality in the given solve time.

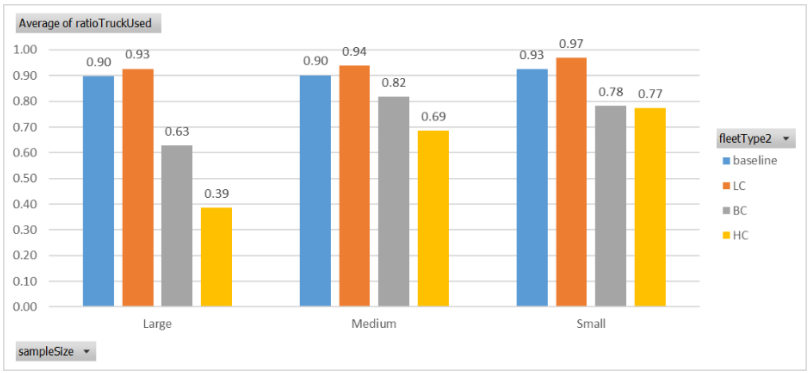


Figure 2. Average fleet utilization versus sample size and fleet type

The results so far indicate that the complexity of the problem leads to lower SR and fleet utilization for the HC fleet. The question becomes then about the advantage of having a heterogeneous fleet and combining multiple trips into one trip, and the advantage is the cost. Our results show that the proposed model achieves lower costs for performing transport requests when high capacity and balanced capacity fleet are employed in any of the sample sizes.

The average cost per request (the ratio of the trip-related component of the objective function to the number of requests satisfied) is plotted against the sample size and fleet type in Figure 3. In each sample size, the lowest cost is achieved by the high and balanced capacity fleets. As expected, the LC fleet performs the requests with the highest cost, as the possibility of combining trips is much less. Our baseline, homogenous fleet of 40-ft trucks, performs better than the low capacity fleet, as the LC fleet includes 20-ft and 40-ft trucks and have less possibility to combine trips compared to a fleet of all 40-ft trucks.

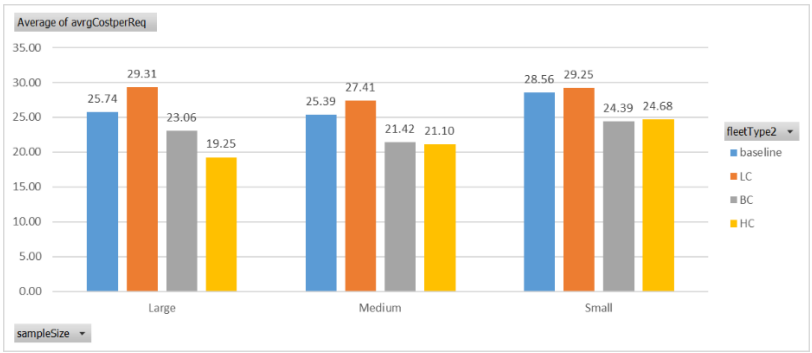


Figure 3. Average cost per transport request versus sample size and fleet type

5 Conclusion

The proposed model relaxes the commonly used simplifying assumptions in the optimization models for drayage operations. The clear advantage is that the optimization model operates in a setting applicable to real world scenarios compared to existing approaches. Our experiments show that, despite the complexity of the problem, our formulation can efficiently allocate container transport requests to trucks for small and medium problem sizes. The results also show that a heterogeneous fleet consisting of high capacity or balanced capacity trucks can achieve lower costs for transport requests by combining

multiple requests into one trip, compared to a homogenous fleet. Due to the complexity of the problem, the future work needs to address an exact or heuristic method to solve the proposed model, rather than solely relying on existing commercial solvers. Such a method would improve success rate and fleet utilization for large samples, as well as high and balanced capacity fleets.

References

[1] C. Macharis and Y. . Bontekoning, "Opportunities for OR in intermodal freight transport research: A review," *Eur. J. Oper. Res.*, vol. 153, no. 2, pp. 400–416, Mar. 2004.

[2] T. E. Notteboom and J.-P. Rodrigue, "Port regionalization: towards a new phase in port development," *Marit. Policy Manag.*, vol. 32, no. 3, pp. 297–313, Jul. 2005.

[3] K. Braekers, A. Caris, and G. K. Janssens, "Integrated planning of loaded and empty container movements," *OR Spectr.*, vol. 35, no. 2, pp. 457–478, Mar. 2013.

[4] G. Giuliano and T. O'Brien, "Reducing port-related truck emissions: The terminal gate appointment system at the Ports of Los Angeles and Long Beach," *Transp. Res. Part D Transp. Environ.*, vol. 12, no. 7, pp. 460–473, Oct. 2007.

[5] F. Schulte, R. G. González, and S. Voß, "Reducing Port-Related Truck Emissions: Coordinated Truck Appointments to Reduce Empty Truck Trips," in *Computational Logistics. Lecture Notes in Computer Science*, N. R. Corman F., Voß S., Ed. Springer, Cham, 2015, pp. 495–509.

[6] Y. Song, J. Zhang, Z. Liang, and C. Ye, "An exact algorithm for the container drayage problem under a separation mode," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 106, pp. 231–254, Oct. 2017.

[7] K. Braekers, A. Caris, and G. K. Janssens, "Integrated planning of loaded and empty container movements," *OR Spectr.*, vol. 35, no. 2, pp. 457–478, Mar. 2013.

[8] X. Wang and A. C. Regan, "Local truckload pickup and delivery with hard time window constraints," *Transp. Res. Part B Methodol.*, vol. 36, no. 2, pp. 97–112, Feb. 2002.

[9] J. Funke and H. Kopfer, "A model for a multi-size inland container transportation problem," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 89, pp. 70–85, May 2016.

[10] R. Zhang, W. Y. Yun, and H. Kopfer, "Multi-size container transportation by truck: modeling and optimization," *Flex. Serv. Manuf. J.*, vol. 27, no. 2–3, pp. 403–430, Sep. 2015.

[11] M. Gendreau, J. Nossack, and E. Pesch, "Mathematical formulations for a 1-full-truckload pickup-and-delivery problem," *Eur. J. Oper. Res.*, vol. 242, no. 3, pp. 1008–1016, May 2015.

[12] D. Popović, M. Vidović, and M. Nikolić, "The Variable Neighborhood Search Heuristic for the Containers Drayage Problem with Time Windows," Springer, Cham, 2014, pp. 351–364.

[13] Z. Xue, C. Zhang, W.-H. Lin, L. Miao, and P. Yang, "A tabu search heuristic for the local container drayage problem under a new operation mode," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 62, pp. 136–150, Feb. 2014.

[14] M. Lai, T. G. Crainic, M. Di Francesco, and P. Zuddas, "An heuristic search for the routing of heterogeneous trucks with single and double container loads," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 56, pp. 108–118, Sep. 2013.

Incentive-Compatible Mechanisms for Traffic Intersection Auctions with Autonomous Vehicles

David Rey
School of Civil and Environmental Engineering
UNSW Sydney
Email: d.rey@unsw.edu.au

Vinayak V. Dixit
School of Civil and Environmental Engineering
UNSW Sydney

Michael W. Levin
Department of Civil, Environmental, and Geo-Engineering
University of Minnesota

1 Introduction

The forecasted emergence of Autonomous Vehicles (AVs) in urban traffic networks provides new opportunities to reduce urban congestion. In particular, AVs are expected to be able to communicate with traffic intersection managers to coordinate movements through intersection. This paradigm has been first proposed by [2] who introduced reservation-based traffic intersection control. In this paper, we study a traffic intersection control problem wherein vehicles have the possibility to indicate their preferences over a set of outcomes. This can be modeled as a combinatorial auction mechanism wherein AVs bid for intersection access and the coordination of traffic is influenced by these bids. A desirable property of mechanism design in the context of public goods is *incentive-compatibility*. That is, the traffic coordination mechanism should incentivise agents to be *truthful* and reveal their true preferences when bidding. A mechanism is said to be Incentive-Compatible (IC) if every agent can achieve the best outcome to itself just by acting according to its true preferences [4]. Our goal is to design an IC mechanism for traffic intersection control assuming that AVs are allowed to bid for intersection access upon arriving at the intersection. The motivation for such a mechanism design is to optimise social welfare by providing agents with the opportunity to influence the system by truthfully reporting their value of time.

Few auction-based mechanisms have been proposed for traffic intersection control [5, 6, 1]. These mechanisms discussed the notion of incentive compatibility and vehicle blocking effects induced by First-In-First-Out (FIFO) constraints at lane queues. Yet, a limitation of the proposed mechanisms is that agents can only participate in the auction if they are either at the front of the queue or if all agents in front of them have been assigned a reservation. Recently, [3] developed a conflict-point formulation to optimize AVs trajectories throughout an intersection. We build on this work and extend the conflict-point model introduced therein for maximizing social welfare in a traffic intersection auction context. By explicitly controlling AVs trajectories, the conflict-point model can account for vehicle blocking effects when considering the bids of participating agents and evaluating each outcome of the auction. We contribute to the field by proposing a Vickrey-Clarke-Groves (VCG) mechanism—which are known to be IC—for AVs traffic control at intersections. Our mechanism takes the form of a combinatorial auction wherein outcomes of the auction are determined by a Mixed-Integer Linear Program (MILP) that maximizes social welfare based on agents' bids. This mechanism is IC and we explore its behavior in a multi-period setting wherein agents have the possibility to participate in multiple auctions until they are able to traverse the intersection. In particular, we discuss fairness considerations and identify refinements of the basic mechanism to prevent low-bidding vehicles to remain delayed for arbitrarily long periods of time.

2 Traffic Intersection Auction Model

We consider a network intersection with pre-defined, possibly conflicting, vehicles movements. We assume that AVs communicate with the intersection manager upon entering a lane queue to access the intersection. The intersection manager is in charge of coordinating vehicle movements through the intersection by providing them with an intersection entry time and a speed. In a congested scenario, AVs may need to wait before traversing the intersection due to conflicting vehicle movements. Hence, we assume that AVs have the possibility to participate in an auction and bid for augmenting their chances to traverse the intersection as soon as possible. The outcome of the proposed traffic intersection auction is a vector of binary values indicating which vehicles will travel through the intersection at the next time period. These binary values are determined by solving the proposed conflict-point MILP which explicitly models all participating agents time-space trajectory through the intersection.

Formally, let $\mathcal{V}$ be the set of vehicles that can bid in the auction with $|\mathcal{V}| = n$. Let $\mathcal{X}$ be the set of outcomes of the traffic intersection auction. For each outcome $\boldsymbol{x} \in \mathcal{X}$, let $z_v(\boldsymbol{x})$ be a binary variable which takes value 1 if vehicle v traverses the intersection in outcome $\boldsymbol{x}$ and 0 otherwise, *i.e.* $\mathcal{X}$ is a set of n-dimensional binary vectors. Let b_v be the bid of vehicle $v \in \mathcal{V}$, we assume that the value function of each agent $v \in \mathcal{V}$ is $b_v z_v(\boldsymbol{x})$. Using this value function, the outcome $\boldsymbol{x}^\star$

maximizing social welfare is:

$$\boldsymbol{x}^\star \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{v\in\mathcal{V}} b_v z_v(\boldsymbol{x}) \tag{1}$$

To determine the payoff of each agent involved in the auction, we adopt a traditional VCG mechanism together with Clarke's Pivot Rule [4]. Specifically, let π_v be the payoff of vehicle $v \in \mathcal{V}$, *i.e.* the amount that v is charged after the auction, the payoffs of vehicle v is:

$$\pi_v = \max_{\boldsymbol{x}\in\mathcal{X}} \sum_{v'\in\mathcal{V}\setminus\{v\}} b_{v'} z_{v'}(\boldsymbol{x}) - \sum_{v'\in\mathcal{V}\setminus\{v\}} b_{v'} z_{v'}(\boldsymbol{x}^\star) \tag{2}$$

Computing all payoff values $\boldsymbol{\pi}$ requires the resolution of $n+1$ optimization problems: one problem involving all n agents and n problems involving all but one agent. Each optimization problem seeks the optimal coordination of agents based on their bids subject to conflict-free traffic conditions. We adapt the conflict-point formulation from [3] to find the outcome $\boldsymbol{x}^\star$ maximizing social welfare as defined by (1).

The conflict-point formulation is a MILP wherein the trajectory of each agent is modeled explicitly and collision avoidance constraints are imposed at each conflict point in the intersection. We assume that the path of each vehicle through the intersection is known (exogenous route choice) and we denote $\boldsymbol{p}_v = (\gamma_v^-, \ldots, \gamma_v^+)$ the path of v corresponding to a sequence of conflict-points starting from γ_v^- and ending at γ_v^+. Let $t_v(c) \geq 0$ (resp. $\tau_v(c)$) be the arrival (resp. reservation) time of v at conflict point $c \in \boldsymbol{p}_v$. Collision avoidance at conflict points are handled using binary variables $\delta_{vv'}(c)$ to model the ordering of vehicles v and v' at conflict point c. Each value of the vector of decision variables $(\boldsymbol{t}, \boldsymbol{\tau}, \boldsymbol{\delta})$ corresponds to an outcome $\boldsymbol{x} \in \mathcal{X}$ of the traffic intersection auction. Hence the set of outcomes $\mathcal{X}$ implicitly represents the feasible region of conflict-point MILP presented in [3].

We model the proposed traffic intersection auction for the time period $[t, t+\Delta t]$ where t is the earliest departure time of any vehicle and Δt represents the period of time over which vehicle trajectories are optimized. We redefine $z_v(\boldsymbol{x})$ as a binary variable indicating if vehicle v traverses the intersection or not at the time period $[t, t+\Delta t]$. Variable $z_v(\boldsymbol{x})$ can be adjusted using the linear constraint (3) wherein M_v is a large enough constant. Hence $z_v(\boldsymbol{x})$ takes value 1 if $t_v(\gamma_v^+) + \tau_v(\gamma_v^+) \leq t + \Delta t$ and is free otherwise.

$$t_v(\gamma_v^+) + \tau_v(\gamma_v^+) \leq t + \Delta t + (1 - z_v(\boldsymbol{x}))M_v \quad \forall v \in \mathcal{V} \tag{3}$$

The MILP formulation for the proposed social welfare traffic intersection auction consists of variables $\boldsymbol{t}, \boldsymbol{\tau}, \boldsymbol{\delta}$ and $\boldsymbol{z}$, the feasible region of the conflict-point MILP presented in [3] together with the constraints (3) and the objective function (1). We refer the reader to [3] for more details on the conflict-point model.

The proposed MILP formulation maximizes social welfare among all agents participating in the auction. To determine the payoff π_v of each agent v, we need only to modify the objective function

(1) by removing vehicle v from $\mathcal{V}$ and solve the corresponding modified MILP. Hence the proposed auction-based mechanism can be implemented as follows:

1. Identify the set of participating agents $\mathcal{V}$, *i.e.* AVs that can traverse the intersection within the time period $[t, t + \Delta t]$
2. Find $\boldsymbol{x}^\star$ by solving the proposed MILP with all agents $v \in \mathcal{V}$
3. Calculate the payoff vector $\boldsymbol{\pi}$ by solving n MILPs, each with one of the n AVs removed from the bidders' set $\mathcal{V}$

The resulting payoff vector $\boldsymbol{p}$ is IC, hence participating AVs are incentivised to truthfully report their value of time. Further, Clarke's Pivot Rule ensures that agents are not forced to bid and that the mechanism does not need to pay the bidders since $\boldsymbol{p}_v \geq 0$. The *revenue* of the mechanism is the sum of the agents' payoffs, *i.e.* $R = \sum_{v \in \mathcal{V}} \boldsymbol{p}_v$. The revenue R may be used in subsequent time periods to subsidize agents with either a low value of time or a low travel budget to avoid the situation of "starvation" (agents waiting an arbitrarily large amount of time) discussed in [5]. We will explore to which extent re-allocating the mechanism revenue among low-bidding agents at ulterior time periods can be used to promote fairness while remaining an IC mechanism.

References

[1] Carlino, D., Boyles, S. D., and Stone, P. Auction-based autonomous intersection management. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (2013), IEEE, pp. 529–534.

[2] Dresner, K., and Stone, P. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2* (2004), IEEE Computer Society, pp. 530–537.

[3] Levin, M. W., and Rey, D. Conflict-point formulation of intersection control for autonomous vehicles. *Transportation Research Part C: Emerging Technologies 85* (2017), 528–547.

[4] Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. *Algorithmic game theory*. Cambridge University Press, 2007.

[5] Schepperle, H., and Böhm, K. Agent-based traffic control using auctions. In *Cooperative Information Agents XI*. Springer, 2007, pp. 119–133.

[6] Vasirani, M., and Ossowski, S. A market-inspired approach for intersection management in urban road traffic networks. *Journal of Artificial Intelligence Research* (2012), 621–659.

Integrated robust & possibilistic multiobjective humanitarian logistic model with social costs

Cristián E. Cortés
Civil Engineering Department
Universidad de Chile, Santiago, Chile

Pablo A. Rey
Industry Department,
UTEM, Santiago, Chile

Luis E. Yáñez
Industrial Engineering Department
Universidad Federico Santa María, Valparaíso, Chile
Email: luis.yanez@usm.cl

1 Introduction

We propose a scenario-based stochastic model for integrated emergency preparedness and response planning for the distribution of emergency supplies after a disaster or catastrophic event. This model extends to a robust possibilist-stochastic version, which incorporates the risk associated with the choice of the expected values to the objective function, as well as penalties for material convergence at the demand points and violations to flow balance constraints.

The model minimizes the social cost associated with deprivation and waiting, identifying a set of possible supply points, where flows are consolidated and sent to predefined facilities, in addition to considering purchasing decisions or prior and subsequent emergency acquisitions. Uncertainty is considered basing our approach in an stochastic programming model based on a set of scenarios constructed from historical occurrence of emergency events. Epistemic uncertainty present in model's parameters is included in the modellig by using fuzzy numbers as values for some of these parameters.

2 Stochastic model of location, allocation and distribution

The distribution network is composed by *aggregated demand points* (ADP) $i \in N$ and *supply points* (SP) $j \in M$. At SPs is possible also to make pre-positioning and post-emergency supply decisions including contributions from external zones to the occurrence of the emergency. ADPs are served with supplies pre-positioned and with flows of supplies received from SPs. On surplus, ADPs can also collaborate with each other. Uncertainty is included in the demand, transport costs and deprivation associated with the time that elapses until the demand at ADPs is satisfied.

The mathematical formulation considers several scenarios of predefined disasters $s \in S$ (each with a probability θ_s of occurrence, determined based on historical records of similar events in the area), including also possible network's damages, as well as deprivation costs Π^s_{ji}. The robust formulation is constructed considering the deterministic costs manifested in phases prior to an emergency, such as the activation or location of supply nodes, purchases or acquisitions and transportation for the prepositioning plus the expected value and a variability measure of the total cost.

Costs associated with post-emergency phases are dependent on different scenarios. We split the post-emergency cost in two parts: $\xi^{(1)}_s, \xi^{(2)}_s$. The first part, $\xi^{(1)}_s$ includes the traditional cost considered in humanitarian logistics associated with purchases or acquisitions, transport and inventory management. The second objective $\xi^{(2)}_s$ includes opportunity costs and inter-temporal effects of human suffering, characterized by deprivation costs calibrated in previous studies [1].

The objective function results then a wighted sum of the three terms corresponding to the deterministic pre-emergency cost and the two parts of post-emergency cost:

$$\Omega = \sum_{j \in M} (F_j Y_j + P^0_j) + \sum_{i \in N} F_i x_i + \sum_{j \in M} \sum_{i \in N} TC^0_{ji} T^0_{ji} q^0_{ji}$$

$$\xi^{(1)}_s = \sum_{k \in V} \sum_{i \in N} \Theta_s TC^s_{ki} T^s_{ki} q^s_{ki} + \sum_{j \in M} \sum_{i \in N} \Theta_s TC^s_{ji} T^s_{ji} o^s_{ji} + \sum_{j \in M} \Theta_s P^s_j p^s_j + \sum_{k \in V} \Theta_s H_k I^s_k$$

$$\xi^{(2)}_s = \sum_{i \in N} \Theta_s W^s_i u^s_i + (\sum_{k \in N} \sum_{i \in N} \Theta_s \pi^s_{ki} q^s_{ki} + \sum_{j \in M} \sum_{i \in N} \Theta_s \pi^s_{ji} q^s_{ji} + \sum_{j \in M} \sum_{i \in N} \Theta_s \pi^{so}_{ki} o^s_{ji})$$

Here, F_j and F_i represent fixed costs for supplying and locating points for prepositioning (SP_j) and receiving help (ADP_i) and Y_j, and X_i, are the corresponding activation binary variables, respectively. TC^0_{ji} and P^0_j represent transport and procurement costs for pre-positioning and pre-emergency procurement, while the variables q^0_{ji} and p^0_j represents associated flows (with known travel distances and given by T^0_{ji}) and pre-disaster purchases.

Scenarios $s \in S$ are previously enumerated and characterized by transport costs (pre-settled or subsequently acquired stock, with distances given by T^s_{ki}, T^s_{ji}, respectively), post-emergency supply costs and maintenance of inventories, given by $TC^s_{ki}, TC^s_{ji}, H_k$ $(k \in V = N \cup M)$. For each scenario, post-emergency decision variables are defined for the existing supply flows or subsequently acquired (q^s_{ki}, o^s_{ji}) and inventory available at the demand nodes I^s_k.

Finally, $\pi^s_{ki}, \pi^s_{ji}, \pi^{so}_{ki}$ denote deprivation costs when serving the node $i \in N$ with flow coming from supply nodes, near demand nodes or post-emergency purchases (corresponding flow variables are $q^s_{ki}, q^s_{ji}, o^s_{ji}$). Unattended demand or shortage in scenario s is represented by the variable u^s_i and the corresponding deprivation cost is W^s_i.

The optimization modelo includes the following constraints:

$$p_j^0 \leq c_j Y_j, \forall j \in M \quad (1)$$

$$p_j^0 \geq v_j^0 = \sum_{i \in N} q_{ji}^0, \forall j \in M \quad (2)$$

$$q_i^0 = \sum_{j \in M} q_{ji}^0 \leq c_i X_i, \forall i \in N \quad (3)$$

$$\varphi_i^s q_i^0 - a_i^s + b_i^s = D_i^s, \forall i \in N, s \in S \quad (4)$$

$$a_i^s \leq |\varphi_i^s q_i^0 - D_i^s|, \forall i \in N, s \in S \quad (5)$$

$$b_i^s \leq |D_i^s - \varphi_i^s q_i^0|, \forall i \in N, s \in S \quad (6)$$

$$q_{ik}^s \leq a_i^s, \forall i, k \in N, s \in S \quad (7)$$

$$\sum_{i \in N} q_{ji}^s = p_j^0 - v_j^0 - I_j^s, \forall j \in M, s \in S \quad (8)$$

$$p_j^s \leq o_j^s Y_j, \forall j \in M, s \in S \quad (9)$$

$$q_{ni}^s \leq b_i^s, \forall n \in V, i \in N, s \in S \quad (10)$$

$$o_{ji}^s \leq b_i^s, \forall j \in M, i \in N, s \in S \quad (11)$$

$$a_i^s - b_i^s - \sum_{n \in N} q_{in}^s + \sum_{n \in N} q_{ni}^s + \sum_{j \in M} (o_{ji}^s + q_{ji}^s) = I_i^s - \mu_i^s, \forall i \in N, s \in S \quad (12)$$

$$Y_j, X_i \in \{0, 1\}, \forall j \in M, i \in N \quad (13)$$

$$q_{ji}^0, q_i^0, v_j^0 \geq 0, \forall j \in M, i \in N \quad (14)$$

$$q_{ni}^s, I_n^s \geq 0, \forall n \in V, s \in S \quad (15)$$

$$a_i^s, b_i^s, \mu_i^s, o_{ji}^s \geq 0, \forall i \in N, j \in M, s \in S \quad (16)$$

Constraint (1) limits pre-emergency acquisition for each SP_j, (2) and (3) restrict stock prepositioning for SP_j, ADP_i. (4) represents equilibrium equation in ADP_i (post-emergency scenarios), while (5) and (6) determine available amount a_i^s or required b_i^s in AD_i[1]. Constraint (7) limits the flow that can be sent from one ADP to another for each post-emergency scenario. Equation (8) represents flow equilibrium for each SP_j, while constraint (9) limits capacity in SP for post-emergency acquisitions. (10) and (11) are capacity constraints on the ADP that receive deliveries from SP or ADP. Finally, (12) defines flow and inventory balance for ADP_i, where u_i^s represents shortage or lack of inventory in ADP_i for a scenario $s \in S$. Constraints (13) to (16) indicate the nature of the variables.

2.1 Robust and multi-objective formulation

The robustness of the solutions is considered by minimizing the absolute value of the variability in post-emergency costs.

The final objective funcion considered in the model then becomes:

$$\min \Omega + \sum_{s \in S} \xi_s^{(1)} + \lambda \sum_{s \in S} \theta_s |\xi_s^{(1)} - \sum_{s' \in S} \theta_{s'} \xi_{s'}^{(1)}| + \sum_{s \in S} \xi_s^{(2)} + \lambda \sum_{s \in S} \theta_s |\xi_s^{(2)} - \sum_{s' \in S} \theta_{s'} \xi_{s'}^{(2)}|$$

Finally, we use the *fuzzy chance constrained programming* approach to deal with the uncertainty on some parameters. Constraints using fuzzy numbers became *possibilistic chance constraints* within selected confidence levels, in order to provide a adequate reliability for the satisfaction of these restrictions. It should be noted that the use of the credibility measure guarantees the satisfaction of the possibilistic objective function and limits the level of certainty according to the modeler's preferences [4]. In the current formulation, parameters T_{ki}^s, T_{ji}^s corresponding to post-emergency travel times are modeled as triangular fuzzy numbers.

[1] φ_i^s represents proportion of the resources prepositioned at $ADPi \in N$ that remains usable in scenario $s \in S$

2.2 Preliminary Results

Taking as a case study the scenarios and instances of [2], [3], the proposed model manages to reduce global costs as the robust and possibilist versions are incorporated, allowing the decision maker to sensitize results to face new disaster scenarios (obtained from fieldwork), as well as to evaluate their impact on the design of public emergency management policies.

References

[1] J. Holguin-Veras, J. Amaya-Leal, V. Cantillo, L.N. Van Wassenhove, F. Aros-Vera, y M. Jaller. Econometric estimation of deprivation cost functions: A contingent valuation experiment. *Journal of Operations Management*, 45:44-56, 2016.

[2] R. Pradhananga, F. Mutlu, S. Pokharel, J. Holguin-Veras, y D. Seth. An integrated resource allocation and distribution model for pre-disaster planning. *Computers and Industrial Engineering*, 91:229-238, 2016.

[3] A. Bozorgi-Amiri, M.S. Jabalameli, y S. M. J. Mirzapour Al-e-Hashem. A multiobjective robust stochastic programming model for disaster relief logistics under uncertainty. *Journal OR Spectrum*. 35(4): 905-933, 2011.

[4] S. Tofighi, S. Torabi y S. Mansouri. Humanitarian logistics network design under mixed uncertainty . *European Journal of Operational Research*. 250(1): 239-250, 2016. Tofighi

Sending a reliable cost-efficient flow through a stochastic time-varying network

Tao Lu
Department of Technology and Operations Management
Erasmus University Rotterdam

Clemens Thielen
Department of Mathematics
University of Kaiserslautern

Rob Zuidwijk
Department of Technology and Operations Management
Erasmus University Rotterdam

Corresponding author: Alberto Giudici
Department of Technology and Operations Management
Erasmus University Rotterdam, Rotterdam, The Netherlands
Email: giudici@rsm.nl

1 Introduction

Traveling as quickly as possibly from a given origin to a given destination in a probabilistic network may require an adaptive routing strategy rather than choosing only a single path (see, e.g., [2]). Consequently, routing decisions that adapt to the probabilistic nature of the network are required in order to minimize expected arrival time. This has already been observed in many different contexts, e.g., for the problem of arriving on time in a bus transport network [3].

In the case of freight transport, where several units of goods need to be sent through a network, network flows are an important and well-studied modelling tool. Real-world freight transport networks, however, are inherently stochastic (e.g., regarding the travel times of trucks or trains on single parts of a route) and the available connections in the network usually vary over time. As the reliability of deliveries as well as the cost of operations are becoming main concerns for transport operators, this provides a strong motivation for studying the computation of reliable, cost-efficient

flows in stochastic, time-varying networks. To the best of our knowledge, this problem has not been studied so far.

For the context of hinterland container transport, we define a model to address the problem of determining the value of adaptively routing flow in a capacitated network with stochastic, time-varying arrival times. The goal is to minimize the total costs incurred for reserving capacity on services prior to the execution of the adaptive plan while reaching a given minimum level of reliability.

Container transport in the hinterland is operated on transport means having different characteristics, chiefly in terms of per unit costs, capacity, and speed. An integrated deployment of those modes led to intermodal transport solutions, where a deterministic planning approach is usually applied. As flows increase and shippers require more reliable transportation, transport operators need to take stochasticity into account directly during their planning. One recently proposed solution is that of synchromodal transport, which extends the concept of intermodal transport by allowing for adaptive decisions about mode and route choice [4, 5]. Our work is motivated by the need to study the value of such a planning approach in order to understand its viability in practice.

2 Model definition

Let $G = (V, R)$ be a directed graph, where the nodes in V represent locations and the arcs in R represent transport services. Let $s, t \in V$ be the source node and the sink node, respectively, and let $K \in \mathbb{N}$ be the number of containers to be routed from s to t earlier than a given deadline $T \in \mathcal{T}$, where $\mathcal{T} = \{t_0, t_1, \ldots, t_l\}$ is the discretized time horizon. Let $c_r \in \mathbb{N}$ be the per unit cost for booking capacity on an arc $r \in R$. We define time-dependent upper capacities $u_{r,\theta} \in \mathbb{N}_{\geq 0}$ ($r \in R$, $\theta \in \mathcal{T}$). For $r \in R$ and $\theta \in \mathcal{T}$ with $u_{r,\theta} > 0$, we let $\tau_{r,\theta}$ be a $\mathcal{T}$-valued random variable representing the arrival time of service r departing at time θ. We let $\kappa \geq 0$ denote the desired minimum expected number of containers being delivered earlier than time T.

Our aim is to select integer capacities $\vec{x} = (x_{r,\theta})_{r \in R, \theta \in \mathcal{T}}$ such that $0 \leq x_{r,\theta} \leq u_{r,\theta}$ for all $r \in R$, $\theta \in \mathcal{T}$ in such a way that the total cost $\sum_{r \in R, \theta \in \mathcal{T}} c_r x_{r,\theta}$ is minimized, while guaranteeing that an optimal adaptive routing strategy using only the booked capacities $\vec{x}$ can deliver at least κ containers earlier than time T in expectation.

By selecting a suitable state space $\mathcal{S}(\vec{x})$, action space $\mathcal{A}(\vec{x})$, reward function $\mathrm{Rew}(\cdot)$, transition probabilities, and strategy space $\Pi^{\mathrm{MD}}(\vec{x})$ depending on the capacity selection $\vec{x}$, the adaptive problem can be formulated as a Markov Decision Process (MDP). The overall optimization model can then be formulated as follows:

$$
\begin{aligned}
&\min \sum_{r\in R}\sum_{\theta\in\mathcal{T}} c_r x_{r,\theta} \\
&\text{s.t. } 0 \le x_{r,\theta} \le u_{r,\theta} \qquad \forall r \in R, \forall \theta \in \mathcal{T} \\
&\quad x_{r,\theta} \in \mathbb{N}_{\ge 0} \qquad \forall r \in R, \forall \theta \in \mathcal{T} \\
&\quad y \ge \kappa \\
&\quad y = \max_{\pi\in\Pi^{\mathrm{MD}}} \mathbb{E}\left[\sum_{i=0}^{T-1} \mathrm{Rew}(X_i, d_i(X_i)) + \mathrm{Rew}^T(X_T) | X_0 = \underline{s}\right] \qquad (1) \\
&\qquad \text{s.t. } \mathcal{S} = \mathcal{S}(\vec{x}) \qquad (2) \\
&\qquad\quad \mathcal{A} = \mathcal{A}(\vec{x}) \qquad (3) \\
&\qquad\quad \Pi^{\mathrm{MD}} = \Pi^{\mathrm{MD}}(\vec{x}). \qquad (4)
\end{aligned}
$$

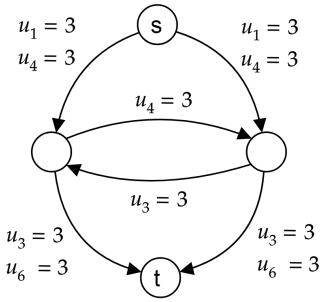
3 Solution approach

We use two different approaches for solving the problem. A first method performs a neighborhood search on the arc capacities solving the MDP for each capacity vector. An optimal solution of the MDP is constructed using backward induction. As only arrival times are stochastic, it is possible to speed up the computations by considering only state transactions having positive probabilities. The second approach follows results of [1], which allowed for a mixed integer programming (MIP) formulation for the whole problem, thus being able to rewrite the whole MDP.

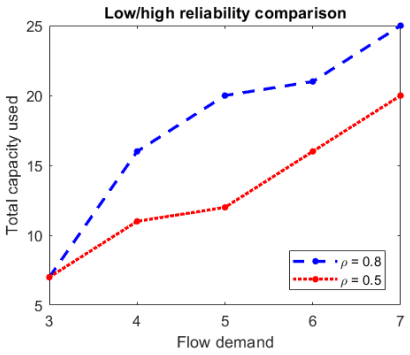
Preliminary results are shown in Figure 1b for the simple graph given in Figure 1a. For each service, there is a positive probability of missing the connection with a following one. Per unit costs have been set to $c_r = 1$ for all $r \in R$. We compare the cost of sending an increasing amount of flow through the network for two levels of reliability $\rho := \frac{\kappa}{K}$ ($\rho_H = 0.8$ and $\rho_L = 0.5$) by choosing $\kappa_\circ = K\rho_\circ$ according to the amount K of flow. Our results show that the additional amount of capacity required to maintain the same level of reliability depends on the reliability itself, but also on the amount of flow. This is in contrast to a deterministic minimum cost flow solution, where sending one additional unit of flow only requires 2 units of additional capacity in total.

4 Conclusion

Our preliminary results focus on the capacity allocation required to send an adaptive flow meeting a certain performance measure. We compare the effect of different levels of reliability on the marginal cost for sending an additional unit of flow. This is done to understand the difference in the planning approach required when including stochastic elements. In a deterministic setting, allocating single units of flow to the cheapest available remaining path in a network is appropriate



(a) Graph instance for Fig. 1b



(b) Cost as a function of the amount of flow K

as minimum cost flows can be decomposed into paths. Our next goal is to study the structure of solutions in the stochastic setting and to investigate whether a flow decomposition into simpler structures (not necessarily paths) similar to the deterministic case is still possible in the stochastic scenario. The results of this formal investigation will support the dialogue with planners who currently operate ad-hoc at a path level to allocate containers on running services. Moreover, we have already discussed our model with practitioners, which led to several possible extensions of the model that will be considered in future research. For example, multiple container sizes could be studied, which might also make it necessary to include stowage at intermediate nodes into the model. Moreover, in addition to uncertainty regarding arrival times, uncertainty may also appear with respect to the departure time of services, which could be integrated into the model.

References

[1] A. Bhattacharya and J.P. Kharoufeh. Linear Programming Formulation for Non-stationary, Finite-horizon Markov Decision Process Models. *Operations Research Letters*, 45(6):570–574, 2017.

[2] R.W. Hall. The Fastest Path through a Network with Random Time-Dependent Travel Times. *Transportation Science*, 20(3):182–189, 1986.

[3] T. Rambha, S.D. Boyles, and S.T. Waller. Adaptive Transit Routing in Stochastic Time-Dependent Networks. *Transportation Science*, 50(3):1043–1059, 2016.

[4] V. Reis. Should we keep on renaming a +35-year-old baby? *Journal of Transport Geography*, 46:173–179, 2015.

[5] B. van Riessen, R.R. Negenborn, and R. Dekker. Synchromodal Container Transportation: An Overview of Current Topics and Research Opportunities. In *Corman F., Vo S., Negenborn R. (eds) Computational Logistics. ICCL 2015.* 2015.

Analytical Delay Models for Interrupted Mixed Flow of Autonomous and Human-Driven Vehicles

Reza Mohajerpoor
The University of Sydney, School of Civil Engineering
Mohsen Ramezani
The University of Sydney, School of Civil Engineering
Sydney, Australia
Email: mohsen.ramezani@sydney.edu.au

1 Introduction and Motivation

Presence of autonomous vehicles (AVs) affects traffic flow characteristics of a mixed traffic stream comprising human-driven vehicles. In particular, AVs can increase the saturation flow of arterials and freeways. As such, studies have been carried out to investigate the effects of AVs in combination with conventional human-driven vehicles (or normal vehicles (NVs)) on the road networks [1, 2]. The main aims of the research in this area are partly to understand the characteristics of the mixed traffic [1] and partly to propose new algorithms to incorporate the real-time information and benefits from connected and automated vehicles (CAV) to improve the mobility of the traffic network [1, 2]. Since AVs have not yet been commercialized, there are numerous aspects that need to be scrutinized regarding the impacts of AVs on road users (i.e. human factors), as well as the traffic flow characteristics of the urban roads and freeways.

To model this impact, we propose an analytical model to derive the expected value of the total delay of a two-lane interrupted road serving a traffic stream with mixed AVs and conventional human-driven (or normal) vehicles (NVs), given the *Expected Penetration Rate* (EPR) of AVs. The models are based on the previously established saturation flow estimator (see [3]) for various possible lane management policies: (a) dedicated lanes, and (b) mixed-mixed lanes. Microsimulation studies demonstrate the validity of the developed delay models.

2 Delay Analysis for a Two-lane Interrupted Road with Mixed Traffic

Total vehicle delay experienced by vehicles stopped behind a traffic light, is a crucial criteria to measure the efficiency of traffic control plans on an arterial road. Let us represent the signal cycle length, and the green time, red time, and loss time of the approach by C, G, R, and L, respectively. It is clear that $C = R + L + G$. Number of vehicles that should be served by the approach in one cycle is thus $n_a = \lfloor q^{\mathrm{a}} C \rfloor$, where q^{a} is the arrival flow of traffic at the controlled approach. We assume the arrival flow is bounded in a way that the approach remains *undersaturated*.

2.1 Dedicated Lanes Policy

In this policy, one lane is dedicated to AVs and one lane is dedicated to NVs. Since the number of AVs arriving during a cycle is a random variable, we use a a binomial distribution to derive the delay relationships. The expected vehicle delay at the approach assuming EPR of p with the dedicated lapolicy can be formulated as [4]:

$$\mathrm{E}[D^{\text{nv-av}}(k, n_{\mathrm{a}})] = \sum_{k=0}^{n_a} D_k^{\text{nv-av}} P(X = k), \tag{1}$$

where

$$D_k^{\text{nv-av}} = \sum_{\zeta=\text{nv,av}} \beta_k^{\zeta} \frac{q_k^{\mathrm{a},\zeta} k^{\mathrm{j}}}{k^{\mathrm{j}} - k_k^{\mathrm{a},\zeta}} \left(R + L_\zeta\right)^2, \tag{2}$$

$$\beta_k^{\zeta} = \frac{k_k^{\mathrm{c},\zeta} \left(k^{\mathrm{j}} - k_k^{\mathrm{a},\zeta}\right)}{k^{\mathrm{j}} \left(k_k^{\mathrm{c},\zeta} - k_k^{\mathrm{a},\zeta}\right)}, \tag{3}$$

k is the number of AVs arrived during the cycle, and k^{j} is the jam density per lane. Moreover, $q_k^{\mathrm{a,av}} = \frac{k}{n_a} q^{\mathrm{a}}$, $q_k^{\mathrm{a,nv}} = \left(1 - \frac{k}{n_a}\right) q^{\mathrm{a}}$, $D_k^{\text{nv-av}}$, $k_k^{\mathrm{a},\zeta}$, and $k_k^{\mathrm{c},\zeta}$ respectively denote the arrival flow of AVs, the arrival flow of NVs, the total vehicle delay of the approach, and the arrival and saturation densities for vehicles of mode $\zeta = \{\text{av}, \text{nv}\}$, given k AVs arrived during the cycle. Concretely, independent from the penetration rate k/n_{a}, the saturation flow for each dedicated lane can be calculated as

$$q_k^{\mathrm{c,av}} = q^{\mathrm{c,av}} = \frac{1}{h_{\text{av}-\text{av}}}, \tag{4}$$

$$q_k^{\mathrm{c,nv}} = q^{\mathrm{c,nv}} = \frac{1}{h_{\text{nv}-\text{nv}}}, \tag{5}$$

where $h_{\text{av}-\text{av}}$ is the headway of an AV following another AV, and $h_{\text{nv}-\text{nv}}$ is the headway of an NV following another NV. To add, given the free flow speed of the upstream traffic, v, the arrival and saturation densities can be directly calculated as follows ($\zeta = \{\text{av}, \text{nv}, \text{m}\}$):

$$k_k^{\text{a},\zeta} = v q_k^{\text{a},\zeta}, \tag{6}$$

$$k_k^{\text{c},\zeta} = v q_k^{\text{c},\zeta}. \tag{7}$$

2.2 Mixed-Mixed Lanes Policy

In the mixed-mixed lanes policy, each lane could have the mixture of AVs and NVs. Intuitively, under this policy user equilibrium dictates that AVs get distributed almost equally among the two lanes. Hence, we make this assumption and later justify it via microsimulation experiments. Accordingly, the expected vehicle delay of the approach reads as

$$\text{E}[D^{\text{m-m}}(k, n_{\text{a}})] = \sum_{k=0}^{n_a} D_k^{\text{m-m}} P(X = k), \tag{8}$$

where

$$D_k^{\text{m-m}} = \beta_k^{\text{m}} \frac{q^{\text{a,m}} k^{\text{j}}}{k^{\text{j}} - k^{\text{a,m}}} (R + L)^2, \tag{9}$$

$q^{\text{a,m}} = 0.5 q^{\text{a}}$, and $k^{\text{a,m}} = 0.5 k^{\text{a}}$. Moreover, the saturation flow of each lane is obtained from the formula below, and the saturation density can be read from (7):

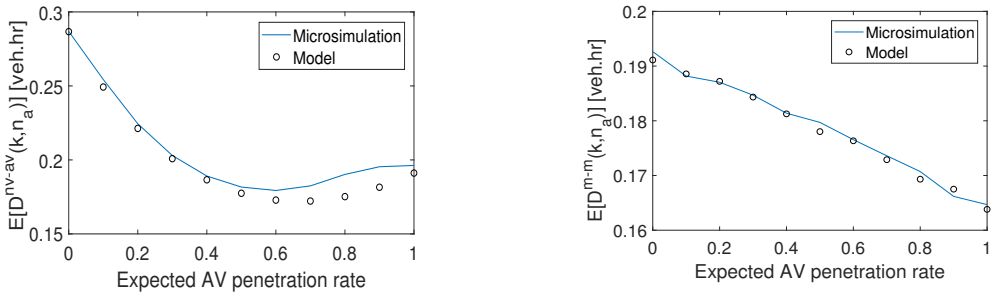
$$q_k^{\text{c,m}} = \frac{1}{\bar{h}_k^{\text{m}-\text{m}}}, \tag{10}$$

where $\bar{h}_k^{\text{m}-\text{m}} = \frac{1}{n_{\text{a}}-1} A_k(n_{\text{a}}) H / C_{n_{\text{a}}}^k$, $H = [h_{\text{nv}-\text{nv}}, h_{\text{av}-\text{av}}, h_{\text{nv}-\text{av}}, h_{\text{av}-\text{nv}}]^T$, $h_{\text{nv}-\text{av}}$ is the headway of an AV following an NV, $h_{\text{av}-\text{nv}}$ is the headway of an NV following an AV, $C_{n_{\text{a}}}^k$ is the number of combinations of k AVs in a platoon of n_a vehicles, and $A_k(\cdot)$ is the $(k+1)$th row of matrix $A(\cdot) \in \mathbb{R}^{(n+1)\times 4}$, which is defined in [3].

2.3 Validation of the Delay Models Using Microsimulation Studies

The evaluation is valuable due to the possible effects coming from the following assumptions (or simplifications) made to develop the delay models: [I] the stochasticity of the headway components (h_{ij}, $i, j \in \{\text{nv}, \text{av}\}$) is ignored, [II] the arrival flow rate is assumed to be constant, [III] under the mixed-mixed lane policy AVs are equally distributed between the lanes, and [IV] the average delay of each lane is approximated by the kinematic wave model [4]. Note that the saturation flow models have been formerly validated via microsimulations in [3].

We conducted microsimulation studies (10 replications of an 1 hr experiment) on a hypothetical intersection, and measured the total vehicle delay at a two-lane approach for various EPRs and different lane management policies. Given that the microsimulation model relaxes the above-mentioned simplifications [I-IV] to mimic the actual behavior of the traffic, one can consider the average delay obtained from the microsimulation model as the ground truth. The results of the microsimulations and their comparison with the proposed models are given in Fig. 1. According to the figure, the modelled delays are close to the microsimulation results for each studied policy.



(a) (b)

Figure 1: Comparison of the expected total vehicle delays obtained from the proposed models and microsimulation studies for various EPRs and different lane management policies, conducted on a two-lane approach.

References

[1] Ghiasi A, Hussain O, Qian Z, Li X. A mixed traffic capacity analysis and lane management model for connected automated vehicles: A Markov chain method. Transportation Research Part B: Methodological. 2017;106:266–292.

[2] Stern RE, Cui S, Delle Monache ML, Bhadani R, Bunting M, Churchill M, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. Transportation Research Part C: Emerging Technologies. 2018;89:205–221.

[3] Ramezani M, Machado JA, Skabardonis A, Geroliminis N. Capacity and delay analysis of arterials with mixed autonomous and human-driven vehicles. In: 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS); 2017. p. 280–284.

[4] Mohajerpoor R, Saberi M, Ramezani M. Delay Variability Optimization Using Shockwave Theory at an Undersaturated Intersection. IFAC-PapersOnLine. 2017;50(1):5289 – 5294. 20th IFAC World Congress.

Fleet sizing and operations management in wildfire suppression operations

Simon Dunstall
CSIRO Data61, Melbourne, Australia
Email: simon.dunstall@csiro.au

Nicholas Davey
Department of Mechanical Engineering
University of Melbourne, Australia

Carolyn Huston
CSIRO Data61, Melbourne, Australia

Edmundo Claro-Rodriguez
CSIRO Land & Water, Santiago, Chile

Saman Halgamuge
Department of Mechanical Engineering
University of Melbourne, Australia

1 Introduction

Interesting classes of transport problem concern emergency response vehicle fleets, and these are generalizable to "mainstream" transport problems. As key examples, we can be seeking to determine the sizes and base locations of air and/or terrestrial vehicle fleets, understand patterns and variability in demand for emergency response services, or dynamically allocate vehicles to bases and/or standby locations in near real-time. Our focus here is on wildfires in grasslands and forests, and is largely based on a case study of a forestry company in Chile. For firefighting this company uses a mixed fleet of helicopters (used for fire crew transport and some water carrying), light aircraft (*Air Tractors*, [1], used for smaller-scale water bombing), and terrestrial brigades. This company has primary firefighting responsibilities through approximately a third of the country by surface area.

Wherever, and whenever, there is significant grass fire or forest fire risk, fire suppression in the initial two hours after ignition is critical. This is often referred to as the *initial attack* phase. Overall wildfire damage minimization outcomes for a fire season hinge on the responsible organisation responding rapidly and containing as many fires as possible during the initial attack phase, especially during periods of more dangerous fire activity (high winds, high ambient temperatures, low relative humidity and/or forest and grassland fuels that are dry). For example, a statistical analyses of Australian wildfires [2,3] show that the fire size encountered at the beginning of fire combat, which is minimised by response rapidity, is one of two important variables for predicting the probability of a wildfire burning 100 ha or more. The other important variable is the severity of fire behaviour due to meteorology and fuel dryness conditions.

Here, we consider an *air fleet sizing problem* (AFSP) and a *dynamic air fleet repositioning problem* (DAFRP). These are founded on data analysis regarding fire occurrence, as well as an evaluation of data on the relative success of initial attack responses which differ in the mix of air and terrestrial vehicles used, and on the time of entry into combat of these brigades. The AFSP we address using statistics and data analysis, whereas the DAFRP is an operational transport fleet optimization problem we address using a hybridization of linear programming and either Model Predictive Control (MPC) or Real Options Valuation (ROV) using Monte-Carlo simulation and control randomization [4].

2 The air fleet sizing problem

Wildfire causes vary, but in many places including Chile and much of Australia, human intentional or negligent causes are in the majority [5,6]. By contrast, in Southern Australia and parts of the United States, lighting strikes especially in mountainous areas are a major cause and can lead to a large number of fires initiated at roughly the same time [7,8]. Occurrence rates have a high spatial variability, with complex dependencies on environmental and social factors. There is a strong diurnal profile in occurrences, with peak times-of-day being specific to classes of fire cause.

Day-to-day variability in fire occurrences is explained partly by ambient temperature, wind speed and humidity. More extreme values of these variables are indicative of higher numbers of (detected) fires, as well as fires that are more difficult to suppress and are more likely to lead to major damage. Except for lightning storm events, the rate of fire occurrence per unit time, or the time between events, is well represented as a random stochastic process.

Our approach is based on statistical simulation that builds upon data analysis regarding fire occurrence. We estimate the maximum observed initial attack demand on the firefighting system, expressed in terms of the number of initial attacks needing to be addressed during a two hour period (for our Chilean case, where intentional and negligent causes predominate). The statistical simulation makes use of a region-by-region estimation of the time between fire ignitions, and has a dependency on time-of-day as well as a forest fire danger index. Monte-Carlo simulations of fire seasons are carried out, and statistics are gathered concerning the initial attack demand in each of three zones: North, Central and South. The result is a quantified cumulative probability distribution where the “likely maximum” number of new fires in a two-hour period across a fire season is taken as the near-100% percentile for each zone. The value of this is nine fires for the busiest zone, and does not exceed twelve fires in total across the company’s territory as a whole in any two hour window. Further consideration of flight distances and the likely maximum in each zone has been used in practice to recommend that a total of 13 pairs of firefighting aircraft and helicopters are on duty each day of the fire season (a force increase of around a third). For the first year that this fleet was in place, the total area impacted by fire was the lowest in a decade, and was less than 5% of the impact in the preceding year.

3 The dynamic air fleet repositioning problem

The DAFRP involves optimally updating the assignments of a fixed number of helicopters and air tractors to bases and to active fires, at intervals over the course of a single day (noting that aerial firefighting is generally not safe at night). These relocations are performed so as to minimise the total expected fire damage, which accumulates from existing fires as well as new fires that might start over

the remainder of the day and which will require initial attack response if significant damage is to be avoided. There is also an operational requirement that no aircraft can exceed its maximum flying hours for the day.

The main operational decisions are around whether to relocate aircraft from their current locations to different locations where there are established fires (that are already beyond being suppressed in initial attack) and/or locations where there are too few aircraft present relative to the risk of new fires (where risk is taken as likelihood multiplied by consequence). There is consumption of airframe flying hours associated with a relocation, trade-offs between the total risk at the origin and destination, and the partial loss of firefighting capacity while the aircraft is repositioning. The risk consideration is most important. We are required to compare the risks posed by existing and potential new fires in different regions (which becomes complex in a stochastic environment where there are existing fires underway), and to avoid wasteful "chasing" of fire risk around the country over the course of a day as a result of risk hotspots moving spatio-temporally due to (predictable) meteorological variation as well as diurnal fire occurrence patterns in each region.

To solve the DAFRP we use a Mixed Integer Linear Programming (MIP) formulation either within an MPC rolling-horizon approach, or use MIP in combination with ROV techniques. The MIP at the centre of the DAFRP assigns aircraft to bases, and aircraft to fires, through binary variables. The MIP objective function has, in effect, two components: one associated with the benefits of assigning aircraft to existing fires and reducing the damage caused by them, and one associated with the benefits of aircraft being present in areas where new fires are more likely. In the MPC approach, there is a fixed proportionality between these objective function components. In the ROV approach, the control decisions (for each hour over the remainder of daylight hours) involve adjusting the weighting between the objective function components. As such, the controls act as decisions at a higher level that the aircraft assignments in the MIP, and can compensate for undue myopia (too much relocation) or inertia (too little relocation) that can otherwise stem from the MIP harnessed within a deterministic MPC approach.

The overall state of the system at a time t we represent as a tuple of three vectors: (i) a measure of the combined severities of burning fires in each region; (ii) the fire behaviour danger in each region, measured by an index; (iii) the cumulative flying hours for each aircraft on the day. There are 16 regions in the problem instances used for developing the algorithms. The expected wildfire damage $D(t)$ for the remainder of the day ("damage to go"), is estimated by a Bellman equation which is recursive over time and incorporates system states and the values of decision variables of the MIP solved at time t. The overall DAFRP objective at time t is to minimise the expected value of $D(t)$.

In the simpler MPC approach, we are limited to a rolling horizon where just one particular realisation of fire occurrence and propagation is considered as the horizon moves forwards. In the ROV approach we estimate $D(t)$ using a large number of Monte Carlo simulations in which fire occurrence times and locations vary between every simulation run. Within each run, the MIP is solved at each future timestep (i.e., hour subsequent to time t) and the control values are also randomized. The results of the large number of simulation runs are regressed so as to build the estimate of $D(t)$ as well as the approximately-optimal control settings. The MIP is nested within the ROV, and even when we smartly decide (algorithmically) when execution of the MIP can be avoided, the approach has an extreme computational expense and requires high performance computing (the nesting is reversed in the road design problem addressed in [9], leading to more moderate computational demands).

The main difference between the capabilities of the MPC and ROV approaches is that in ROV we optimize against a large number of possible future realisations at every timestep, whereas the MPC accounts only for expected values and a single deterministic fire scenario that unfolds over time. The ROV should therefore be much more robust to variability in future firefighting demands (which are spatio-temporally predictable but highly variable), and give better long-run (day after day) performance avoiding "catastrophes" when more extreme events occur.

4 Conclusion

The MPC approach to the DAFRP problem shows how MIP-based optimisation can be applied to dynamically reposition a heterogenous multi-asset fleet so as to maximise emergency response effectiveness. The ROV approach to DAFRP demonstrates that the hybridization of real options and combinatorial optimization is feasible not just for strategic planning as in [9], but for near real-time applications of decision-making under uncertainty. This we believe is an exciting and under-explored intersection of research into combinatorial optimisation (OR), financial mathematics and stochastic control, and by necessity involves the harnessing of high performance computing resources.

References

[1] Air Tractor, https://airtractor.com/, accessed 21 October 2018

[2] M. Plucinski, Factors Affecting Containment Area and Time of Australian Forest Fires Featuring Aerial Suppression, *Forest Science* 58(4), 390–398, 2012

[3] M. Plucinski, Modelling the probability of Australian grassfires escaping initial attack to aid deployment decisions, *International Journal of Wildland Fire* 22, 459–468, 2013

[4] I. Kharroubi, N. Langrene and H. Pham, A numerical algorithm for fully nonlinear HJB equations: An approach by control randomization, *Monte Carlo Methods and Applications* 20, 145–165, 2014

[5] M. Plucinski, W.L. McCaw, J.S. Gould and B.M. Wotton, Predicting the number of daily human-caused bushfires to assist suppression planning in south-west Western Australia, *Int. J Wildland Fire* 23, 520–531, 2014

[6] M. Plucinski, The timing of vegetation fire occurrence in a human landscape, *Fire Safety Journal* 67, 42–52, 2014.

[7] C. Miller, M. Plucinski, A. Sullivan, A. Stephenson, C. Huston, K. Charman, M. Prakash and S. Dunstall, Electrically caused wildfires in Victoria, Australia are over-represented when fire danger is elevated, *Landscape and Urban Planning* 167, 267–274, 2017

[8] S.L. Stephens, Forest fire causes and extent on United States Forest Service lands, *Int. J Wildland Fire*, 14, 213–222, 2014

[9] N. Davey, S. Dunstall and S. Halgamuge, Optimal road design through ecologically sensitive areas considering animal migration dynamics, *Transport Research C*, 478–494, 2017

Pickup and delivery problem with truckload synchronization through multiple cross-docks

Yousef Maknoon

Faculty of Technology, Policy and Management,

Delft University of Technology, Delft, Netherlands

Gilbert Laporte

Canada Research Chair in Distribution Management,

HEC Montral,Montral, Canada

Email: M.Y.Maknoon@tudelft.nl

1 Introduction

This paper presents an adaptive large neighborhood search algorithm for the pickup and delivery problem with the choice of truckload synchronisation through multiple cross-docks. The problem is to find a set of minimum-cost routes as well as the proper vehicle load synchronization at cross-docks to satisfy a set of pickup and delivery requests. In the distribution network, cross-docks act as freight consolidation points to achieve economies of scale.

All studies in this problem consider a network with a single cross-dock and the main optimization concern is to synchronize the pickup and delivery routes (See [1],[2] and [3]). In this paper, we extend the problem and studied the general network with multiple cross-docks. Similar to all studies in routing problem with cross-docking, we consider two separate routes (pickup and delivery) for products transportation. However, in each route the transferred products can be consolidated with other products through transshipment operations at multiple cross-docks.

2 Problem description

A logistic company operates with multiple cross-dock facilities. The company is responsible for transferring a set of n customer requests ($r \in R$, $|R| = n$). Requests can pass through multiple cross-docks to be consolidated with other requests. We identify each request $r \in R$ with three attributes : 1) demand load, 2) pickup location and 3) delivery location. The transportation

process is divided into two separate shifts (pickup and delivery), where each shift has a maximum working time. The company uses identical vehicles ($k \in K$) of capacity Q. All vehicles start and end their route at their assigned cross-dock. Moreover, multiple visits of the same cross-dock during the same shift are forbidden. The problem is defined on a directed graph. With each arc in this graph, we associate two non-negative values: a travel distance and a service time. The scheduling problem involves decisions on route design and consolidation at cross-docks in order to minimize the total travel distance.

In order to model vehicle load synchronization at the cross-docks, we represent the transportation network by means of two planning levels. At the first level, we focus on the flow of products moving from one vehicle to another for the entire planning horizon (pickup and delivery shifts) as well as on the vehicle routes between cross-docks. We decompose vehicle route into segments. Each segment has its origin and destination cross-docks, and no cross-dock can be visited in between. Moreover, for each segment, we define three modes to transport requests. The mode determine whether the vehicle will pick up, deliver or haul the request. For each request a binary variable is used to decide about its transport mode using vehicle k on its route between cross-docks.

The first-level decisions are then linked to those at the second level where explicit routing decisions have to be made for each vehicle. We use the three-index binary variable to model vehicle route. Additional variables are also used to control the maximum travel time. The objective is to find a set of minimum-cost vehicle routes to serve all requests.

3 Solution approach

We have devised an adaptive large neighborhood search (ALNS) heuristic. The algorithm has three main components: 1) destroy operators to remove a set of requests from the solution, 2) repair operators to re-insert the destroyed requests into the solution, and 3) an adaptive mechanism to allocate the search time among the operators, based on their historical performance and also to control the intensification and diversification of the search.

The value of solution S is computed as $F(S) = Z(S) + \xi_1 L(S) + \xi_2 T(S)$, where $Z(S)$ is total travelled distance, $L(S)$ denotes the total capacity violation (the load that exceeds the vehicle capacity), and $T(S)$ represents the total violation of working time. In our implementation, we allow infeasible solutions and penalize total violations by means of parameters ξ_1 and ξ_2 which are dynamically adjusted during the search process.

At each iteration the algorithm modifies the current solution by applying a destroy operator and a repair operator. The newly constructed solutions are then evaluated and are possibly accepted according to a criterion. The search process terminates after a given number of iterations.

We have developed eight operators to destroy a solution. Some of these operators are adopted

from those proposed by Pisinger and Ropke [4]. The destroy operators seek to modify the current solution S by removing requests from it.

We have also developed three repair operators: cluster insertion, best route insertion and regret insertion. The cluster insertion operator follows a cluster-first route-second strategy. The key idea behind this approach is to cluster request locations according to their geographical distance from cross-docks. The best route insertion follows the same request selection criteria as the cluster insertion. However, the operator gives the highest priority to the routes with minimum insertion cost. Finally, the regret insertion use a look-ahead strategy on selecting requests. For each location, the regret value is defined as the difference between inserting it in the best and the second best routes. Similar to the best route insertion, this operator gives the highest priority to the route regret value.

4 Computational experiments

The algorithm was coded in C++ and we chose the solution of our mathematical formulation as a benchmark to evaluate the performance of the algorithm.

We evaluate the performance of the ALNS algorithm on a set of instances derived from the one proposed by Solomon [5], which differ according to the distribution of nodes: random (R), clustered (C) and a mixture of cluster and random (RC). For each type, we have considered 18 settings by varying the number of available cross-docks, the number of vehicles and the number of requests. Finally, we have generated four instances for each case and overall 72 instances are tested. We ran the algorithm five times for each instance and compared the best and the average solution values with that of the solution obtained by running CPLEX with a time limit of 24 hours.

Table 1 summarizes the computational results on the C, R and RC instances. For each group of instances we report the best (Max Best Imp.) and the mean (Mean Average Imp.) percentage improvement of the ALNS solution value with respect to the best CPLEX solution. The average ALNS solution time in seconds is reported under "Mean Time (seconds)". The headings "Min Gap LB" and "Max Gap LB" report the minimum and the maximum percentage difference of the best ALNS solution value with respect to the best lower bound found by CPLEX.

For most of the instances, the average solution value obtained by ALNS over five runs is very close to that obtained by CPLEX. However, the best solution reported by the ALNS is better than that obtained by CPLEX. This improvement is considerable for instances with a larger number of requests. For most of the instances the ALNS solution value is around 5% higher than the best lower bound returned by CPLEX.

Table 1: Summary of computational results, two cross-docks

		$\|K\| = 4$, $Q = 500$			$\|K\| = 6$, $Q = 300$			$\|K\| = 8$, $Q = 250$		
Number of requests (n)		**15**	**30**	**50**	**15**	**30**	**50**	**15**	**30**	**50**
C	**Max Best Imp.**	0.00%	0.00%	0.22%	0.00%	0.00%	1.00%	0.00%	0.00%	2.26%
	Mean Average Imp.	-0.47%	-0.76%	-1.22%	0.00%	-1.16%	-0.13%	0.00%	-1.23%	-0.79%
	Mean Time (seconds)	71.72	141.70	378.27	61.74	122.18	347.18	63.21	106.38	295.76
	Max Gap LB	1.89%	2.81%	5.24%	0.00%	3.24%	4.57%	0.00%	6.19%	8.17%
	Min Gap LB	0.00%	0.00%	3.59%	0.00%	0.09%	1.72%	0.00%	0.65%	5.00%
R	**Max Best Imp.**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	Mean Average Imp.	0.00%	-0.02%	-2.02%	0.00%	-0.43%	-1.89%	-0.02%	-0.97%	-1.42%
	Mean Time (seconds)	71.25	138.86	411.43	66.99	137.76	300.75	65.05	137.05	192.71
	Max Gap LB	0.00%	0.00%	3.94%	0.00%	2.86%	3.39%	0.00%	2.45%	4.85%
	Min Gap LB	0.00%	0.00%	1.69%	0.00%	0.00%	1.71%	0.00%	0.00%	3.56%
RC	**Max Best Imp.**	0.00%	-0.60%	-0.06%	0.00%	0.00%	1.00%	0.00%	0.00%	1.45%
	Mean Average Imp.	0.00%	-1.93%	-1.45%	-0.02%	-0.68%	-1.05%	-0.33%	-1.84%	-0.75%
	Mean Time (seconds)	62.15	136.14	361.38	71.17	117.55	391.03	60.46	122.68	443.92
	Max Gap LB	0.00%	3.14%	4.23%	0.00%	3.55%	4.25%	0.00%	4.65%	4.72%
	Min Gap LB	0.00%	0.60%	2.12%	0.00%	0.00%	1.68%	0.00%	0.00%	2.54%

5 Conclusions

We have described a pickup and delivery problem in which requests have to be processed at least by one cross-dock. The aim of the problem was to compute a minimum cost routes by synchronizing the vehicle loads via several cross-docks. We have presented a mathematical formulation of the problem and provided an adaptive large neighborhood search algorithm. The results demonstrate that our algorithm can find high quality solutions within reasonable computation time

References

[1] C. Mues, S. Pickl, Transshipment and time windows in vehicle routing, in: 8th International Symposium on Parallel Architectures,Algorithms and Networks (ISPAN'05), IEEE Computer Society, 2005, pp. 113–119.

[2] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau, G. Laporte, Vehicle routing with cross-docking, Journal of the Operational Research Society 60 (12) (2009) 1708–1718.

[3] C. D. Tarantilis, Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking, Optimization Letters 7 (7) (2013) 1583–1596.

[4] D. Pisinger, S. Ropke, A general heuristic for vehicle routing problems, Computers & Operations Research 34 (8) (2007) 2403–2435.

[5] M. M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (2) (1987) 254–265.

Response to Reviewers

Thank you for the careful review of our paper *"Max-Pressure Based Autonomous Intersection Management with Pedestrians"*, and for the thoughtful comments. We carefully considered the comments. The following describes how we addressed the comments (reviewers' comments are in italics):

Reviewer #1

The study addresses a problem with the existing Autonomous Intersection Management methods, the fact that they do not consider the pedestrian flow at the intersections. A modified max-pressure control algorithm is proposed for that to account for the pedestrian flow in determining the optimal throughput for intersections of a network. The algorithm is used to calculate the activation of each turning movement and pedestrian signals at every time step by defining movement weights. The algorithm optimizes the sum of the pressure of all movements. The algorithm is based on the assumption that all pedestrians are served in a single time step. The study is interesting. The mathematics of the problem have been defined rigorously.

1) *What challenges implementation of such algorithm might impose on real practice? What are the solutions for those?*

The first challenge is to get the measured queue length of vehicles at the intersection. It requires all vehicles are equipped with V2I devices so they can transmit their location information to the controller at the intersection. Another challenge is to get the estimated arrival rate of pedestrians, for which we need to know the route choice and the trip distribution of pedestrians.

2) *The input from pedestrian parts of the objective function, can they be gathered in actual practice?*

To calculate pedestrian weights, estimated pedestrian queue length and pedestrian waiting time are needed. Pedestrian time can be directly collected with the timer connected with the press button at the crosswalk and we assume that pedestrians will press the button when they arrive the crosswalk. However, the estimated pedestrian queue length requires the information on pedestrian paths, which is hard to measure in the field.

3) *Can this intersection optimization scheme be administered in practice?*

The algorithm proposed in this study is a distributed intersection optimization algorithm. Each intersection only needs to optimize its own intersection control. A central controller can be utilized to connect with all intersections and collect information from each of the intersection to monitor the performance of the network.

4) *The proposed method to estimate the queue length of pedestrians, how accurate is that, and to what degree the optimization outcome is sensitive to the accuracy of such estimation?*

There is still uncertainty in the variation of pedestrian queues in the network under the control of the max-pressure algorithm. According to the simulation, the difference between the actual pedestrian queue length and the estimated pedestrian queue length is bounded.

We modified the abstract to address question 1 and 2. We are unable to address question 3 and 4 in the abstract because of the page limit, but we will include them in the full paper.

Reviewer #2

The proposed research aims to add consideration of pedestrian flows to existing autonomous traffic control strategies. The mathematical problem formulation is provided in detail and seems plausible. I am looking forward to seeing numerical examples that demonstrate the model.

Thanks. Because of the page limit, we will not put the numerical examples in the extended abstract but we will include numerical examples in our paper on max-pressure intersection control.

Max-Pressure Based Autonomous Intersection Management with Pedestrians

Rongsheng Chen
Department of Civil, Environmental, and Geo- Engineering
University of Minnesota

Jeffrey Hu
Department of Computer Science and Engineering
University of Minnesota

Michael W. Levin
Department of Civil, Environmental, and Geo- Engineering
University of Minnesota
Email: mlevin@umn.edu

David Rey
School of Civil and Environmental Engineering
UNSW Sydney

1 Introduction

Autonomous intersection management (AIM) is an intersection control mechanism in which all vehicles approaching an intersection send their information to the controller of the intersection and follow its instructions. It was first proposed by Dresner and Stone [1]. They modeled an intersection with autonomous vehicles as a multi-agent system and proposed a reservation-based approach which outperformed the traditional intersection control with traffic lights. Hausknecht et al. [2] applied AIM-based agents to control interconnected intersections in a network. Existing studies about AIM show great efficiencies compared with traditional intersections but ignore the pedestrian flow at the intersection. This is a limitation of existing models since in practice pedestrian trajectories cannot be precisely controlled like those of autonomous vehicles. To improve the accessibility of AIM, it is necessary to consider pedestrian flows when designing an AIM algorithm.

The control algorithm proposed in this study is based on max-pressure control algorithms. Max-pressure algorithm allows a network to use decentralized controllers for each of the intersections with the traffic state data from its adjacent intersections, so it uses less time to calculate signal timings compared with centralized controllers (SCOOT and OPAC). Although max-pressure controls have been applied to control intersection signals [3, 4], their formulations do not account

for pedestrian flows. This study proposes a modified max-pressure control based on Varaiya's study [4] and considers pedestrian flows in the network to produce the optimal throughput for each intersection after defining the weight of movements.

2 Network Model

Consider a traffic network consisting of a road network for vehicles $\mathcal{G}(\mathcal{N}, \mathcal{L})$ and a sidewalk network for pedestrians $\mathcal{R}(\mathcal{N}, \mathcal{L})$. For both networks, $\mathcal{N}$ denotes the node set, $\mathcal{L}$ denotes the link set. The link set can be divided into three subsets $\mathcal{L}_{\text{entry}}$, $\mathcal{L}_{\text{inter}}$, $\mathcal{L}_{\text{exit}}$ representing entering, internal, and exiting links respectively. The entering link set includes links that bring vehicles or pedestrians to the road network or the sidewalk network. The exit link set includes links that take vehicles or pedestrians out of the road network or the sidewalk network.

In the road network, a pair of links (i, j) denotes a turning movement leaving link i and entering link j. The capacity of turning movement (i, j) is calculated by $Q_{ij} = \min\{Q_i, Q_j\}$. Let $\mathcal{M}$ be the set of all turning movements. Let Γ_i^- and Γ_i^+ be the sets of incoming and outgoing links of link i respectively. We assume an intersection is divided into several conflict regions where trajectories of turning movements intercept with each other. Let $\mathcal{C}$ be a set for all conflict zones at an intersection. $\mathcal{C}_{ij}$ is the set of conflict zones on the trajectory of turning movement (i, j). The capacity of conflict zone c is Q_c and is determined by $Q_c = \max_{\{(i,j)|c\in\mathcal{C}_{ij}\}}\{Q_{ij}\}$. Let δ_{ij}^c denote the relation between turning movement (i, j) and conflict region c. If turning movement (i, j) intersects with conflict region c, $\delta_{ij}^c = 1$, otherwise, $\delta_{ij}^c = 0$. The activation of turning movement (i, j) is represented by $S_{ij} \in [0, 1]$, which is the percentage of time used for activating turning movement (i, j) in a time step. The number of vehicles allowed to move from i to j is calculated by $y_{ij} = S_{ij}Q_{ij}$. For each conflict region, the sum of S_{ij} should not be larger than 1, because the total time occupied by any turning movement should not be larger than a time step.

In the sidewalk network, two sidewalks are directly connected or connected by a crosswalk. A pair of sidewalks (m, n) can denote a crosswalk which connects sidewalks m and n. Let Γ_m^- and Γ_m^+ be the sets of incoming and outgoing sidewalks of sidewalk m. Let $\mathcal{W}$ be the set of all crosswalks. In some cases, different pairs of sidewalks may represent the same crosswalk when it connects multiple pairs of sidewalks. If vehicle turning movement (i, j) intersects with crosswalk (m, n), $\delta_{ij}^{mn} = 1$, otherwise, $\delta_{ij}^{mn} = 0$. Let Q_{mn} be the maximum service rate of crosswalk (m, n), which is assumed to be larger than the maximum pedestrian arrival rate at crosswalk (m, n). In this study, pedestrian flows are controlled by signals. We assume that all pedestrians can be served in a time step. The activation of crosswalk (m, n) at time t is represented by $\mathcal{Z}_{mn}(t)$. When the pedestrian signal is activated, $\mathcal{Z}_{mn}(t) = 1$, otherwise, $\mathcal{Z}_{mn}(t) = 0$. The actuation of the pedestrian signal is related with the queue length. Let $\tau_{mn}(t)$ be the waiting time of pedestrians from sidewalk m to sidewalk n since the last actuation of the pedestrian signal. $\tau_{mn}(t)$ can be updated with equation (1).

$$\tau_{mn}(t) = \begin{cases} \tau_{mn}(t) + 1, & \tau_{mn}(t) \geq 0 \wedge \mathcal{Z}_{mn}(t) = 0 \\ 1, & \tau_{mn}(t) = 0 \wedge x_{mn}(t) \geq 0 \\ 0, & \mathcal{Z}_{mn}(t) = 1 \vee x_{mn}(t) = 0 \end{cases} \tag{1}$$

To update queue lengths of movements, a point queue model is used, as shown in equation (2). $x_{jk}(t)$ is the queue length of turning movement jk at time t, $y_{jk}(t)$ is the turning flow at time t, and the last term is the total amount of flows that join the queue jk from upstream links. p_{jk} is the proportion of vehicles on link j going to link k. The information on the route choice and the trip distribution are needed to calculate p_{jk}. The flow $y_{ij}(t)$ is calculated at every time step based on the control algorithm.

$$x_{jk}(t+1) = x_{jk}(t) - y_{jk}(t) + \sum_{i \in \Gamma_j^-} y_{ij}(t) p_{jk}(t) \tag{2}$$

2.1 Stability region

Let the demand vector $\mathbf{d}$ represent the traffic entering the network through entering links. Let f_i, f_{ij} denote the total flows on link i and on a turning movement (i, j) respectively. A demand vector $\mathbf{d}$ can uniquely determine a flow pattern on the network. Let $S(t)$ be an intersection control matrix that includes values of $S_{ij}(t)$ for all vehicle turning movements at time step t. $\mathcal{S}$ is an intersection control sequence which includes all $S(t)$ from time step 1 to T. The long-term average time used for activating the turning movement (i, j) can be calculated by getting the average value of $S_{ij}(t)$.

$$\overline{S}_{ij} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} S_{ij}(t) \tag{3}$$

A demand vector $\mathbf{d}$ can be stabilized if there is an intersection control sequence $\mathcal{S}$ that can make the average link flow rate $\overline{f_i}$ and the average serving time $\overline{S}_{ij}$ follow the relation in equation (4) for every link. Let $\mathcal{D}^o$ denote the set of demands that can be stabilized.

$$\overline{f_i}\overline{p}_{ij} \leq \overline{S}_{ij} Q_{ij}, \forall i, j \in \mathcal{L} \tag{4}$$

2.2 Max-pressure control policy

This study uses a max-pressure algorithm to calculate the activation of each turning movement and pedestrian signals at every time step. The weight of each vehicle turning and crosswalk can be calculated by equation (5) and (6) respectively. The weight is the queue length at the upstream link or sidewalk minus the average queue length at downstream links or sidewalks. In equation (7), to calculate the weight of crosswalk (m, n), an estimation of queue length x_{mn}^{ped} is used because it is hard to measure the pedestrian queue length. In equation (7), τ_{mn} is the waiting time since the last actuation of the pedestrian signal. $\overline{u}$ is the mean arrival rate of pedestrians at the crosswalk (m, n) and need to be estimated in advance using the information on the route choice and the trip distribution of pedestrians. We assume that the difference between the estimated queue length and the actual queue length is bounded, which is $|x_{mn}^{\text{ped}}(t) - \overline{u}_{mn}\tau_{mn}| \leq \epsilon$.

$$w_{ij}^{\text{veh}} = x_{ij} - \sum_{k \in \Gamma_j^+} x_{jk} p_{jk} \tag{5}$$

$$w_{mn}^{\text{ped}} = x_{mn}^{\text{ped}} - \sum_{o \in \Gamma_n^+} x_{no}^{\text{ped}} p_{no} \tag{6}$$

$$x_{mn}^{\text{ped}} = \tau_{mn} \overline{u}_{mn} \tag{7}$$

After calculating the weight for each turning or crosswalk, a mathematical program is used to calculate the intersection control strategy, as shown in equation (8).

$$\max \quad \sum_{(i,j) \in \mathcal{M}} w_{ij}^{\text{veh}} y_{ij} + \sum_{(m,n) \in \mathcal{W}} w_{mn}^{\text{ped}} Q_{mn} \mathcal{Z}_{mn} \tag{8a}$$

$$\text{s.t.} \quad y_{ij} \leq Q_{ij}(1 - \mathcal{Z}_{mn} \delta_{ij}^{mn}), \qquad \forall (i,j) \in \mathcal{M}, \forall (m,n) \in \mathcal{W} \tag{8b}$$

$$\sum_{(i,j) \in \mathcal{M}} y_{ij}(t) \delta_{ij}^{c} \leq Q_c, \qquad \forall c \in \mathcal{C} \tag{8c}$$

$$y_{ij} \leq x_{ij}, \qquad \forall (i,j) \in \mathcal{M} \tag{8d}$$

$$Z_{mn} \in \{0,1\}, \qquad \forall (m,n) \in \mathcal{W} \tag{8e}$$

$$y_{ij} \geq 0, \qquad \forall (i.j) \in \mathcal{M} \tag{8f}$$

The max-pressure control aims to optimize the sum of pressures of all turnings and crosswalks. y_{ij} represents the number of cars in turning movement (i,j) that is allowed to move. The values of S_{ij} can be calculate by $S_{ij} = y_{ij}/Q_{ij}$. Let S^* to denote the max-pressure control of all intersections in the network.

Proposition 1. *If the demand vector $\boldsymbol{d} \in \mathcal{D}^o$, this max-pressure control is stabilizing.*

The control method is proved to be stabilizing. The queue length of vehicles and pedestrians in the system is bounded. This control method can also achieve optimal throughput as an optimization model is used to get the control strategy.

References

[1] Dresner, Kurt and Stone, Peter, "A multiagent approach to autonomous intersection management", *Journal of artificial intelligence research* 31, 591–656 (2008).

[2] Hausknecht, Matthew and Au, Tsz-Chiu and Stone, Peter, "Autonomous intersection management: Multi-intersection optimization", *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* 4581–4586 (2011).

[3] Le, Tung and Kovács, Péter and Walton, Neil and Vu, Hai L and Andrew, Lachlan LH and Hoogendoorn, Serge SP, "Decentralized signal control for urban road networks", *Transportation Research Part C: Emerging Technologies* 58, 431–450 (2015).

[4] Varaiya, Pravin, "Max pressure control of a network of signalized intersections", *Transportation Research Part C: Emerging Technologies* 36, 177–195 (2013).

Integer Programming Models for Freight Logistics Service Network Design with In-Tree Constraints

Natashia Boland and **Ira Wheaton Jr.**
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology, USA
Email: natashia.boland@isye.gatech.edu, ira.wheaton@isye.gatech.edu

1 Introduction

The Service Network Design Problem with In-Tree Constraints (SNDPITC) is the problem of finding a minimum-cost transportation plan for shipping multiple less-than-truckload (LTL) commodities from their respective origin to their respective destination. In addition, if any commodities sharing the same destination should meet at any given point in their respective paths, then they must continue on the same path from that point forward. This restriction implies not only that each commodity follows a single path, but that the paths for commodities having the same destination induce an in-tree rooted at the destination.

More formally, let $G = (N, A)$ be a directed graph with node set N and arc set A. Let the cost of a truck traversing arc $a \in A$ be $m_a \in \mathbb{R}_{>0}$. Let K denote the set of all commodities, $o(k), d(k) \in N$ denote the origin and destination for commodity $k \in K$, respectively, and $q(k) \in (0, Q]$ denote its quantity, where Q is the capacity of a truck. In what follows, we take $Q = 1$ (commodity quantities are scaled accordingly). The SNDPITC seeks a shipping plan in which each commodity $k \in K$ follows a single path in G from $o(k)$ to $d(k)$. Furthermore, for any $k_1, k_2 \in K$ such that $d(k_1) = d(k_2)$, if the paths for k_1 and k_2 visit node $n \in N$, then both paths must depart n on the same arc $a \in A$. The cost of the shipping plan, which the SNDPITC seeks to minimize, is modeled by

$$\sum_{a \in A} m_a \Big\lceil \sum_{k \in K : a \in P_k} q(k) \Big\rceil$$

where $P_k \subseteq A$ denotes the arcs used in the path for commodity k.

Service network design (SND) problems in which the flow of a commodity must follow a single path, (it cannot be split to be sent on multiple paths), are known as SND with *unsplittable* or *nonbifurcating* flow, and have been relatively well studied (Frangioni and Gendron, 2009), as have SND problems with various transport cost functions (Fortz et al., 2017). However, LTL companies often, in load planning, require the in-tree restriction

(Powell and Koskosidis, 1992), which is used to simplify operations at handling terminals (crossdock facilities and the like): dock workers loading trucks need only look at the destination of a commodity to know the next destination it should be loaded to. To our knowledge, the SNDPITC has only been studied, to date, by Powell and Koskosidis (1992), who present a local improvement heuristic which manipulates the tree constraint, as well as primal-dual algorithms that provide upper and lower bounds.

Here we formulate and compare alternative integer programming (IP) models of the SNDPITC in terms of their size and strength, both theoretically (Section 2) and computationally (Section 3). We also consider approaches to strengthening the formulations.

2 IP models for the SNDPITC

We give three IP formulations. All use n_a, a non-negative integer variable representing the number of truckloads needed on arc a, for each $a \in A$. Two use binary flow variables to model the path for each commodity: let x_{ak} for arc a in the network be a binary variable indicating whether the arc is used ($x_{ak} = 1$) or not ($x_{ak} = 0$) in the path for commodity k. Our IP Formulation 1 (F1) is given below:

$$\min \quad z = \sum_{a \in A} m_a n_a \tag{1}$$

$$\sum_{a \in \delta^+(i)} x_{ak} - \sum_{a \in \delta^-(i)} x_{ak} = \begin{cases} 1, & \text{if } i = o(k) \\ -1, & \text{if } i = d(k) \\ 0, & \text{otherwise} \end{cases}, \quad \forall i \in N, \forall k \in K \tag{2}$$

$$x_{ak} + \sum_{\substack{a' \in \delta^+(i), \\ a' \neq a}} x_{a'k'} \leq 1 \quad \forall i \in N, \forall a \in \delta^+(i), \forall k, k' \in K(d) \text{ and } k \neq k' \tag{3}$$

$$n_a \geq \sum_{k \in K} q_k x_{ak} \quad \forall a \in A \tag{4}$$

where $\delta^-(i)$ and $\delta^+(i)$ denote, as usual, the set of arcs coming into and going out of node i, respectively, and $K(d)$ denotes the set of all commodities with destination d. Constraint (3) models the in-tree requirement. Integrality constraints are assumed.

Our IP Formulation 2 (F2) introduces the binary variable, y_{ad} indicating whether arc a is used ($y_{ad} = 1$) or not ($y_{ad} = 0$) in the in-tree for commodities with destination d. These variables induce a tree rooted at node d with arcs directed towards the root. F2 uses objective (1) with constraints (2) and (4). The relationship between the x_{ak} and y_{ad} variables, together with the in-tree requirement, is now modeled as

$$x_{ak} \leq y_{ad}, \qquad \forall a \in A, \forall d \in D, \forall k \in K(d) \tag{5}$$

$$\sum_{a \in \delta^+(i)} y_{ad} \leq 1, \qquad \forall i \in N, \forall d \in D \tag{6}$$

where $D \subseteq N$ denotes the set of commodity destination nodes.

Our IP Formulation 3 (F3) introduces the continuous variable $w_{ad} \in \mathbb{R}_{\geq 0}$ indicating the quantity flowing on arc a that is destined for d, which replaces the binary x_{ak} variable for $k \in K(d)$. F3 uses objective (1) with constraints (4) and (6), along with

$$\sum_{a \in \delta^+(i)} w_{ad} - \sum_{a \in \delta^-(i)} w_{ad} = \begin{cases} \sum_{k \in K(d): o(k) = i} q_k, & \text{if } i \neq d \\ -\sum_{k \in K(d)} q_k, & \text{if } i = d \end{cases}, \quad \forall i \in N, \forall d \in D \tag{7}$$

$$w_{ad} \leq \big(\sum_{k \in K(d)} q_k \big) y_{ad}, \quad \forall a \in A, \forall d \in D. \tag{8}$$

The following three simple constraints can be added to strengthen the models:

$$n_a \geq x_{ak}, \ \forall k \in K, \quad y_{ad} \leq \sum_{k \in K(d)} x_{ak} \quad \text{and} \quad \big(\min_{k \in K(d)} q_k \big) y_{ad} \leq w_{ad}, \ \forall d \in D, \quad \forall a \in A.$$

The first applies to all of F1, F2 and F3, the second to F2 only and the third to F3 only.

It is not difficult to see that F1 has $\mathcal{O}(|K||A|)$ variables and $\mathcal{O}(|K|^2|A|)$ constraints, F2 has the same order of number of variables but only has $\mathcal{O}(|K||A|)$ constraints, while F3 has $\mathcal{O}(|N||A|)$ variables and constraints. Hence F3 is the smallest of these formulations. Indeed, since it is generally the case that $|N| << |K|$, F3 is significantly more compact.

We can prove the following result analytically.

Proposition 2.1 *F2 is the strongest of these formulations.*

Thus the large number of constraints required for F1 is not accompanied by an associated formulation strength; the most interesting trade-off is between F2 and F3.

By observing that the in-tree for each destination node is a Steiner tree in a graph, we may exploit Steiner tree concepts, as in (Koch and Martin, 1998), to strengthen F3, in particular. All formulations may be strengthened by cut-set inequalities (Raack et al., 2011), however there are challenges in separating such inequalities (Chouman et al., 2016).

3 Preliminary computational results

We have generated multiple instances of the SNDPITC, using the approach described by Baubaid et al. (2018), with varying parameters. The number of end-of-line (EOL) terminals in the network ranges from 10 to 15, while the number of breakbulk terminals ranges from 2 to 4. The percentage of EOL terminals with demand between them is 50%, so the number of commodities ranges from 45 to 105.

We used the commercial solver Gurobi to attempt to solve each instance with a 30 minute CPU run-time limit. The table shows preliminary results on four instances. The solver reached the run-time limit on almost all instances (instance 1 was solved within 6

Instance and Formulation		No. Variables	No. Constraints	Root Node Gap (%)	Final Gap (%)	No. B&B Nodes	Primal Gap (%)
1	F1	1,932	10,284	6.53%	0.01%	8,009	0.00%
	F2	2,352	3,422	5.98%	0.01%	14,743	0.00%
	F3	882	1,532	9.67%	0.72%	76,382	0.00%
2	F1	2,852	14,022	5.98%	2.30%	11,911	0.00%
	F2	3,472	4,852	4.85%	1.34%	23,367	0.00%
	F3	1,302	2,192	13.96%	5.78%	29,142	0.71%
3	F1	6,572	49,153	3.77%	0.90%	31,241	0.00%
	F2	7,502	10,457	5.13%	0.75%	70,674	0.00%
	F3	1,922	3,347	5.34%	1.36%	50,482	0.04%
4	F1	9,222	69,159	5.84%	4.35%	995	0.59%
	F2	10,527	14,097	5.47%	3.86%	4,175	0.00%
	F3	2,697	4,557	8.21%	4.74%	11,027	0.28%

minutes for F1 and F2). The primal gap reported is the difference between the least cost feasible solution found by the formulation, relative to that found by any formulation.

References

Ahmad Baubaid, Natashia Boland, and Martin Savelsbergh. Dealing with demand uncertainty in service network and load plan design. In WJ van Hoeve, editor, *CPAIOR 2018, Lecture Notes in Computer Science*, volume 10848, pages 63–71. Springer, 2018.

Mervat Chouman, Teodor Gabriel Crainic, and Bernard Gendron. Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667, 2016.

Bernard Fortz, Luís Gouveia, and Martim Joyce-Moniz. Models for the piecewise linear unsplittable multicommodity flow problems. *European Journal of Operational Research*, 261(1):30–42, 2017.

Antonio Frangioni and Bernard Gendron. 0–1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Math.*, 157(6):1229–1241, 2009.

Thorsten Koch and Alexander Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998.

Warren B. Powell and Ioannis A. Koskosidis. Shipment routing algorithms with tree constraints. *Transportation Science*, 26(3):230–245, 1992.

Christian Raack, Arie Koster, Sebastian Orlowski, and Roland Wessäly. On cut-based inequalities for capacitated network design polyhedra. *Networks*, 57(2):141–156, 2011.

Recent Advancements in Solution Methods for Traveling Salesman Problems with a Drone

Mark Bierema
Econometric Institute, Erasmus School of Economics
Erasmus University Rotterdam, The Netherlands

Eveline van Dijck
Econometric Institute, Erasmus School of Economics
Erasmus University Rotterdam, The Netherlands

Paul Bouman
Econometric Institute, Erasmus School of Economics
Erasmus University Rotterdam, The Netherlands
Email: bouman@ese.eur.nl

1 Introduction

In the past decade the inclusion of Unmanned Aerial Vehicles, commonly referred to as drones, into various logistic transportation systems has received increased attention from researchers. It is not difficult to imagine the potential of a vehicle that is not bounded to the road network nor dependent on a driver and which is even able to deliver a parcel on your doorstep or in your back yard. Early concepts proposed by companies such as Amazon suggested that drones can deliver goods directly from the depot to the consumer. While such a system has the advantage that routing becomes trivial as the drones fly directly between depot and customer, the distance covered by a drone will be much greater than the distance covered by a truck performing an optimized vehicle tour. As a consequence, different approaches where drones collaborate with other vehicles or operate within a multi-echelon setup are now being considered. Many of these new approaches give rise to interesting new routing problems.

One of the approaches that was proposed is to have a truck collaborate with a drone. By acting as a mobile depot to the drone, the unique advantages of the drone can be exploited to make deliveries to locations that are far away or difficult to reach for a truck, while the truck can make other deliveries at the same time. This way the total distance covered by the vehicles

remains reasonable, yet the time required in which all deliveries can be performed is decreased substantially due to the parallel operations of both vehicles. In computational experiments with random Euclidean instances where the drone travels twice as fast as the truck, it was observed that 30% of time can be saved compared to an approach where only the truck is used [Agatz et al., 2018]. A theoretical analysis finds that the efficiency improvement is related to the square root of the ratio of the speeds of the truck and drone [Carlsson and Song, 2017]. A generalization where multiple trucks and multiple drones perform the deliveries was considered by Wang et al. [2017] and Poikonen et al. [2017] and the authors provide a number of worst-case bounds.

A first formalization of the routing problem where a single truck collaborates with a single drone was introduced by Murray and Chu [2015] as the *flying sidekick traveling salesman problem* along with a polynomial sized yet relatively weak MIP formulation and a suitable local search heuristic. One notable feature of the *flying sidekick traveling salesman problem* is that it is not allowed for the vehicles to visit the same location twice. As the approach assumes that the drone and truck can only interact with each other at customer locations and not *on the road*, it can be beneficial to allow repeated visits to location as to create more rendez-vouz opportunities for both vehicles. This aspect was included in the *traveling salesman problem with drone* studied by Agatz et al. [2018], which does make the assumption that the vehicles can only interact at customer locations, but does allow for repeated visits. This variant was solved using a MIP formulation with an exponential number of constraints and variables, which turned out to be able to solve instances of up to 10 locations exactly. Furthermore, a number of heuristics which outperform the earlier heuristics were introduced and applied to instances of up to 250 locations. In a follow up work [Bouman et al., 2018] dynamic programming algorithms were developed which allow us to solve instances in the range of 15 to 20 locations exactly.

In this talk, we discuss recent advancements in the development of techniques that can solve larger instances to these problems, and provide insights into the challenges going forward.

2 Column Generation and Cutting Planes

The different approaches introduced by Agatz et al. [2018] rely on the concept of an *operation*, which consists of a start location, and end location, a set of locations visited by the truck and optionally a single drone location that is visited by the drone exclusively. At the start of the operation the drone and truck are together, but in between their paths may vary. The MIP approach can now be summarized as follows: choose a set of operations that visit all locations such that the sum of their costs is minimized, ensuring that if we interpret operations as arcs from their start to end location the operations span a directed Eulerian subgraph. An Eulerian subgraph is generalization of a tour that allows for locations to be visited multiple times and as

a consequence the formulation bears similarities to common formulations for the regular TSP. As a single operation can include any number of locations which are visited by the truck and the formulation introduces a variable for each operation, it is clear that this formulation does not scale very well when the number of locations in the instance increases. To make matters worse, the formulation depends on an exponential number of subtour elimination constraints. In this talk we consider how we can overcome these obstacles using *column generation* and *cutting plane* techniques to generate the variables and subtour elimination constraints on the fly.

To deal with the exponential number of variables in the model, we consider column generation. The pricing problem of this approach consists of finding a new operation that improves the current solution of the master problem. This pricing problem requires us to select a start location, end location and a drone location, and optionally a route visiting any number of truck locations. We investigate both a MIP based approach and a labeling algorithm to solve the pricing problem. We also consider some rules that can be used to prune the set of valid operations to a smaller set which is still guaranteed to contain an optimal solution. In preliminary computational experiments we are able to find optimal solutions to instances with up to 34 locations. Furthermore, the pruning rules for the sets of operations are able to provide considerably reductions in the number of operations that have to be considered. Finally, a heuristic variant of the proposed exact techniques is able to improve upon the solutions obtained by the local search heuristics that were developed earlier.

To deal with the exponential number of subtour elimination constraints, we consider cutting plane procedures. As a first step we have developed a MIP model that can be used to separate the subtour elimination constraints. We find that this procedure already results in a significant improvement of the solution procedure. Furthermore, we have considered whether a useful class of valid inequalities, the comb inequalities [Hong, 1971], can be extended to the traveling salesman problem with drone. While we were unable to find a satisfactory class of valid inequalities that can deal with the situation where locations can have repeated visits, we were able to formulate additional valid inequalities for the setting where repeated visits are forbidden. We propose two separation procedures that can separate subsets of these valid inequalities. The first is an exact MIP-based procedure that is able to separate blossom inequalities. The second is a heuristic separation procedure derived from the procedure by Padberg and Rinaldi [1990]. Furthermore, we include the set of *logical inequalities* that were introduced for the orienteering problem [Leifer and Rosenwein, 1994]. In preliminary computational experiments we find that this last class improves the solution procedure considerably. We also find that the comb based valid inequalities can be violated when no violated subtour elimination constraints exist, yet as this is a relatively rare occurence for the instances we tested it is currently a challenge to show their effectiveness in practice.

3 Discussion and Challenges

Using our proposed methodologies for the traveling salesman problem with drone, we are able to solve considerably larger instances than we were able to solve before. We introduce a column generation procedure that can generate operations dynamically rather than having to generate all operations a priori. Furthermore, we introduce a cutting plane procedure for the subtour elimination constraints and introduce additional valid inequalities and separation procedures for the case where locations can not have repeated visits. This suggests that allowing repeated visits yields a more challenging problem. As a direction for future research, it would be interesting to analyze in which circumstances repeated visits yield better solutions, and how likely these situations occur in practice. One dimension of this question is whether repeated visits are actually necessary in cases where we allow the truck and the drone to interact *on the road*.

References

Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 2018.

John Gunnar Carlsson and Siyuan Song. Coordinated logistics with a truck and a drone. *Management Science*, 2017.

Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697, 2017.

Stefan Poikonen, Xingyin Wang, and Bruce Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017.

Chase C Murray and Amanda G Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.

Paul Bouman, Niels Agatz, and Marie Schmidt. Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 2018.

S. Hong. *A linear programming approach for the travelling salesman problem.* PhD thesis, Johns Hopkins University, Baltimore, Maryland, USA, 1971.

Manfred Padberg and Giovanni Rinaldi. Facet identification for the symmetric traveling salesman polytope. *Mathematical programming*, 47(1-3):219–257, 1990.

Adrienne C Leifer and Moshe B Rosenwein. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*, 73(3):517–523, 1994.

An approach to model competition in ridesharing

Venktesh Pandey
Department of Civil, Architectural, and Environmental Engineering
The University of Texas at Austin
venktesh@utexas.edu

Julien Monteil
Control and Optimization Group
IBM Research, Ireland Lab

Andrea Simonetto
Control and Optimization Group
IBM Research, Ireland Lab

1 Introduction

Dynamic ridesharing and ride hailing has seen a significant increase in the recent past [1]. A growing number of private companies participate in the shared market of providing rides to customers in real-time, including Uber, Lyft, DiDi, and Via, and naturally compete against each other to gain a significant market share. This competition may lead to non-optimal behavior, in terms of the service rate achieved, and of the number of cars present on the roads. In this work, we aim to provide a realistic model of the competition between ride-sharing companies operating at a city-scale, and quantify its impact in terms of the deviation from optimality.

Modeling competition among private entities has been studied in different areas of transportation including network design problems [2]. In a recent work [3], the taxi market is modeled as a multiple leader-follower game and an approximate Nash equilibrium for the competition is solved. However, these models assume a large extent of information sharing among the private companies which is not realistic, like assuming that each company knows the location of each vehicle of every other company. In addition, extending these models to large scale networks and to dynamic settings is a challenge. Recent works [4, 5] provide solutions for solving the centralized real-time city-scale ridesharing problem, by mapping incoming batch of requests with available vehicles, in a three-step procedure: (i) selecting candidate vehicles to serve requests, (ii) computing serving costs meeting ridesharing constraints, and given those computed costs (iii) performing optimal assignments of requests to vehicles. It is highlighted that linear assignments can perform as good as more elaborated assignments, when run at a high enough sampling rate. Hence we propose to evaluate the competition structure between ridesharing companies by modeling ridesharing as batches of linear assignment problems. A system optimal assignment is defined as the one that minimizes the total system cost of assigning any vehicle to a rider, and this goes on for consecutive batches of requests. The main contribution of our work is the modeling of the competition and cooperation scenarios under different realistic settings of information sharing between the ridesharing companies and a

given central authority, e.g. a city authority.

2 Method

The actors of the ridesharing system under consideration are the users requesting the rides via their smartphones in real-time, the multiple companies offering the rides via their available vehicles, and possibly a central agent coordinating the process depending on the level of information shared.

Let $\mathcal{M}$ denote a set of customer trip requests at time t and $\mathcal{P}$ denote a set of competing companies. For a given batch of requests, each company $p \in \mathcal{P}$ has a fleet of available vehicles, denoted by set $\mathcal{C}_p$, which are available for the riders to request. $\mathcal{C}$ denotes the set of all available vehicles, $\mathcal{C} = \bigcup_p \mathcal{C}_p$. Let $x_{ij} \in \{0,1\}, i \in \mathcal{C}, j \in \mathcal{M}$ be the set of binary variables: $x_{ij} = 1$ only if vehicle i is assigned to customer j, otherwise $x_{ij} = 0$. Let $c_{ij} \in \mathbb{R}_+$ denote the cost incurred by vehicle i if it is assigned to customer j. This cost is typically defined as the travel distance or travel time, considering the different vehicle and rider parameters including detour time required to add the new customer to the current schedule, vehicle capacity, preference of customers already on-board etc.

For each new batch of requests, the system optimal assignment is the solution of the linear assignment problem of Equations (1), referred as $\mathbf{LAP}(C, \mathcal{M})$. We assume equal number of riders and vehicles, that is $|\mathcal{C}| = |\mathcal{M}|$, for solving this assignment problem. If $|\mathcal{C}| < |\mathcal{M}|$ (respectively, $|\mathcal{C}| > |\mathcal{M}|$), we can add dummy vehicles (riders) with infinite cost of being assigned to every other rider (vehicle).

$$
\begin{aligned}
\min_{x_{ij} \in \{0,1\}, i \in \mathcal{C}, j \in \mathcal{M}} & \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{M}} c_{ij} x_{ij} \quad [\mathbf{LAP}(C, \mathcal{M})] \\
\text{subject to: } & \sum_{i \in \mathcal{C}} x_{ij} = 1, \quad \forall j \in \mathcal{M} \\
& \sum_{j \in \mathcal{M}} x_{ij} = 1, \quad \forall i \in \mathcal{C} \\
& c_{ij} = \mathcal{I}(i,j), \quad \forall i \in \mathcal{C}, j \in \mathcal{M}
\end{aligned}
\tag{1}
$$

We focus on three protocols of interaction between multiple companies, highlighted in Figure 1. The arrows represent information sharing between the entities. We have:

- A `Centralized` protocol where a centralized agent computes the costs associated with assigning each vehicle to any rider and seeks to find the optimal solution to the $\mathbf{LAP}(C, \mathcal{M})$. This can be solved using efficient and scalable algorithms, e.g. centralized auction algorithms [6].
- A `Distributed` protocol where the central agent does not require access to the location of the vehicles, instead the central agent interacts with each company by iteratively asking for

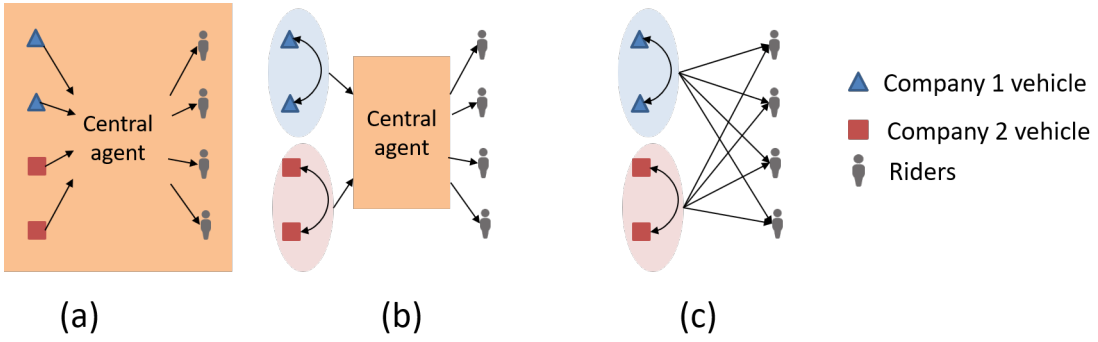


Figure 1: Three different settings for multi-company information sharing. (a) `Centralized` protocol (b) `Distributed` protocol (c) `No-communication` protocol

the two best bid prices for the yet unassigned riders. This protocol, which is essentially an extension of the distributed auction algorithm [7], runs until all riders are assigned to vehicles. It relies on a distributed architecture, and minimizes information sharing (no proprietary information of the companies is shared) but, as we shall prove, still enables to solve $\mathbf{LAP}(\mathcal{C}, \mathcal{M})$ to optimality. Extensions of this protocol are the `Stochastic Distributed` protocol which aims to model the noise that can come from the vehicle positioning errors and the routing computations, and the `Stochastic Distributed with Bias` protocol as the routing/map service may be different among the companies and a given routing service may provide consistently shorter/longer travel times compared to others.

- A `No-communication` protocol where there is no central agent to coordinate the assignment process, and riders send their requests to the company, for example through a broker application on their smartphone. Each company p solves $\mathbf{LAP}(C_p, \mathcal{M})$ and submit bid offers directly to riders j with $x_{ij} = 1, i \in \mathcal{C}_p$. The riders who receive an offer then select the best offer among the companies and are considered assigned. Each company then resolves the assignment considering all unassigned riders. The process continues until all riders are assigned. An extension of this protocol is the `No-communication with Preference` protocol which models the fact that riders always choose the offer from the company they prefer.

Our objective is to compare the performance of the information sharing settings in `Distributed` and `No-communication` protocols against the optimal performance of the `Centralized` protocol.

3 Selected results

We get the following results:

1. We prove that the `Distributed` protocol converges to the optimal assignment and the number

of iterations for convergence is a quadratic function of the number of vehicles. The proof follows closely the one in [7].

2. We prove that with the `No-communication` protocol, the total system cost at convergence is at worst 3 times more than the optimal cost.

3. We test the performance of the protocols on different cost matrices. As an example, Figure 2 shows the optimality gap in % for some randomly generated cost matrices (benefit is defined as the negative of the cost). As observed, in the case of 2 and 3 companies, it is between 0.1–16%.

4. We conduct tests on large scale ridesharing instances, using the NYC Taxi dataset [8]. We show how an increased optimality gap significantly leads to deteriorating the service rate for the protocols under consideration.

Two company scenario				
	3X3 matrix	**50X50 matrix**	**100X100 matrix**	**350X350 matrix**
Optimal assignment benefit:	130	48626.22	20637.67	348444.29
Company 1 no. of vehicles	2	30	48	250
Company 2 no. of vehicles	1	20	52	100
No-communication benefit	120	48089.92	20554.99	347845.03
Percent below than the optimal	7.69%	1.10%	0.40%	0.17%
Three company scenario				
	3X3 matrix	**50X50 matrix**	**100X100 matrix**	**350X350 matrix**
Optimal assignment benefit:	130	48626.22	20637.67	348444.29
Company 1 no. of vehicles	1	16	33	104
Company 2 no. of vehicles	1	21	30	138
Company 3 no. of vehicles	1	13	37	108
No-communication benefit	120	47650.46	20461.74	347894.56
Percent below than the optimal	7.69%	2.01%	0.85%	0.16%

Figure 2: Results for different randomly generated matrices for `No-communication` protocol

The findings of the presented research advocate the presence of a central agent that coordinates the assignment process between the different companies (the `Distributed` protocol). We also draw conclusions on how the results can help design policies that can improve system performance even in the case of pure competition between ridesharing companies (the `No-communication` protocol).

References

[1] Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.

[2] Hua Wang and Xiaoning Zhang. Game theoretical transportation network design among multiple regions. *Annals of Operations Research*, 249(1-2):97–117, 2017.

[3] Xinwu Qian and Satish V Ukkusuri. Taxi market equilibrium with third-party hailing service. *Transportation Research Part B: Methodological*, 100:43–63, 2017.

[4] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[5] Andrea Simonetto, Julien Monteil, and Claudio Gambella. Real-time city-scale ridesharing via linear assignment problems. *Under peer-review*, 2018.

[6] Dimitri P Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, 1990.

[7] Oshri Naparstek and Amir Leshem. Fully distributed optimal channel assignment for open spectrum access. *IEEE Transactions on Signal Processing*, 62(2):283–294, 2014.

[8] NYC Taxi and Limousine Commission dataset. available online.

On the Price of Satisficing in Network User Equilibria

Mahdi Takalloo

Changhyun Kwon

Department of Industrial and Management Science Engineering

University of South Florida

Email: chkwon@usf.edu

1 Introduction

Instead of assuming a perfectly rational person with a clear system of preferences and perfect knowledge of the decision-making environment, we can consider *boundedly* rational persons with (1) an ambiguous system of preferences and (2) lack of complete information, following Simon [6]. When decision makers are indifferent among alternatives within a certain threshold, they are called *satisficing* decision makers [6], opposed to *optimizing* decision makers. Satisficing decision makers choose any alternative whose utility level is above a threshold, called an *aspiration level*, even when the alternative is not optimal.

The satisficing behavior is related to the first source of boundedness—an ambiguous system of preferences. While the travel-time minimization has been traditionally used as a basis for drivers' route-choice modeling, sub-optimal route-choice behavior has also gained attention [1–3], as empirical evidence has supported it [7]. In the literature, a traffic pattern equilibrated among rational drivers is called the perfectly rational user equilibrium (PRUE), while a traffic pattern equilibrated among satisficing drivers is called a boundedly rational user equilibrium (BRUE). We will use *satisficing user equilibrium* (SatUE) instead of BRUE to emphasize that it only considers the ambiguous system of preferences.

The main contribution of this paper is the quantification of how bad the total travel time in SatUE can be, both *analytically* and *numerically*. We introduce the *user equilibrium with perception error* (UE-PE) model to capture SatUE flow vectors. Next, similar to the notion of price of anarchy [5], which compares the performances of the system optimal solutions and the PRUE solutions, we define the *price of satisficing* (PoSat) as the ratio between the worst-case total travel time of SatUE and the total travel time of PRUE. We provide some theoretical bounds for PoSat. We utilize

the sensitivity analysis of parametric variational inequalities [4] for the numerical quantification of PoSat.

2 Notation and Definitions

Let us define the set of path flow variables $\boldsymbol{f}$ and the set of link flow variables $\boldsymbol{v}$ as

$$\mathcal{F} = \left\{ \boldsymbol{f} : \sum_{p \in \mathcal{P}_w} f_p = Q_w \quad \forall w \in \mathcal{W}, \qquad f_p \geq 0 \quad \forall p \in \mathcal{P} \right\}$$

$$\mathcal{V} = \left\{ \boldsymbol{v} : v_a = \sum_{p \in \mathcal{P}} \delta_a^p f_p \quad \forall a \in \mathcal{A}, \qquad \boldsymbol{f} \in \mathcal{F} \right\}$$

where f_p is the flow in path p, $\mathcal{W}$ is the set of OD pairs, $\mathcal{P}_w$ is the set of all paths for OD pair w, $\mathcal{P}$ is the set of all paths, Q_w is the demand for OD pair w, and constant δ_a^p is 1 if link a is on path p and is 0 otherwise. We denote arc travel time function with arc traffic volume of $\boldsymbol{v}$ by $t_a(\boldsymbol{v})$. We denote the travel time function along path p with flow $\boldsymbol{f}$ by $c_p(\boldsymbol{f})$. Moreover, we denote the arc-based total travel time function by $Z(\boldsymbol{v})$ and the path-based total travel time function by $C(\boldsymbol{f})$. Furthermore, we define $\mathcal{F}_{1+\kappa}$ as the set of feasible path flows when the travel demand is increased by the factor of $(1+\kappa)$; that is

$$\mathcal{F}_{1+\kappa} = \left\{ \boldsymbol{f} : \sum_{p \in \mathcal{P}_w} f_p = (1+\kappa) Q_w \quad \forall w \in \mathcal{W}, \qquad f_p \geq 0 \quad \forall p \in \mathcal{P} \right\}.$$

We define the PRUE and SatUE as follows:

Definition 2.1 (Perfectly Rational User Equilibrium) *A traffic pattern $\boldsymbol{f}^0$ is called a* perfectly rational user equilibrium *(PRUE), if*

$$\text{(PRUE)} \qquad f_p^0 > 0 \implies c_p(\boldsymbol{f}^0) = \min_{p' \in \mathcal{P}_w} c_{p'}(\boldsymbol{f}^0) \quad \forall p \in \mathcal{P}_w, w \in \mathcal{W} \tag{1}$$

Definition 2.2 (Satisficing User Equilibrium) *A traffic pattern $\boldsymbol{f}^\kappa$ is called a* satisficing user equilibrium *with κ, or κ-SatUE, if*

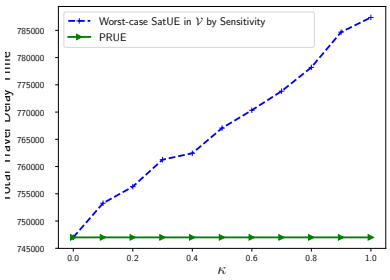
$$\text{(SatUE)} \qquad f_p^\kappa > 0 \implies c_p(\boldsymbol{f}^\kappa) \leq (1+\kappa) \min_{p' \in \mathcal{P}_w} c_{p'}(\boldsymbol{f}^\kappa) \quad \forall p \in \mathcal{P}_w, w \in \mathcal{W} \tag{2}$$

In the context of bounded rationality and satisficing, we are more interested in comparing the total travel time under approximate Nash equilibrium—equivalently SatUE—and the perfectly rational user equilibrium; $C(\boldsymbol{f}^\kappa)$ and $C(\boldsymbol{f}^0)$, respectively. We define the price of satisficing (PoSat) of instance ρ as follows:

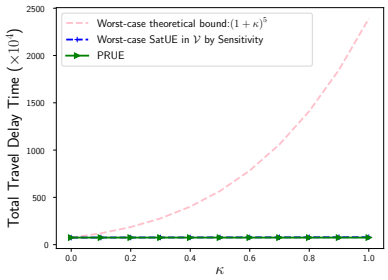
$$\text{PoSat}(\rho) = \max_{\boldsymbol{f}^\kappa \in \Psi_\kappa(\rho)} \frac{C(\boldsymbol{f}^\kappa)}{C(\boldsymbol{f}^0)}, \tag{3}$$

where $\Psi_\kappa(\rho)$ represents the set of SatUE flows in instance ρ.

Related to SatUE, we introduce the user equilibrium with perception error (UE-PE) model. In this model, network users are seeking the shortest path; however, they have their own perception of the travel time function.



(a) Numerical SatUE worst-case total travel time



(b) Theoretical SatUE worst-case total travel time

Figure 1: Comparing the SatUE worst-case total travel time with the PRUE total travel time for the Sioux Falls network

Definition 2.3 (UE-PE-$\mathcal{V}$) *Let ε_a denote the users' perception error of travel time along arc a. A flow link vector $\bar{\boldsymbol{v}} \in \mathcal{V}$ is a solution to the* UE-PE-$\mathcal{V}$ *model, if*

$$\sum_{a\in\mathcal{A}}(t_a(\bar{\boldsymbol{v}}) - \varepsilon_a)(v_a - \bar{v}_a) \geq 0 \qquad \forall \boldsymbol{v} \in \mathcal{V} \tag{4}$$

for some $\varepsilon_a \in [0, \frac{\kappa}{1+\kappa} t_a(\bar{\boldsymbol{v}})]$.

We can show that UE-PE-$\mathcal{V}$ implies SatUE.

Lemma 2.1 (UE-PE-$\mathcal{V}$ $\implies$ SatUE) *Suppose $\bar{\boldsymbol{v}}$ is a solution to* UE-PE-$\mathcal{V}$ *in* (4) *with some $\bar{\boldsymbol{\varepsilon}}$ where $\bar{\varepsilon}_a \in [0, \frac{\kappa}{1+\kappa} t_a(\bar{\boldsymbol{v}})]$ for all $a \in \mathcal{A}$. Then $\bar{\boldsymbol{v}}$ is a κ-SatUE flow.*

3 Theoretical Bounds for PoSat

Using Lemma 2.1, we can provide a bound on the performance of UE-PE flows for a special case:

Theorem 3.1 *When the link travel time functions are in the monomial form of $t_a(v_a) = b_a(v_a)^n$, let $\boldsymbol{v}^\kappa$ be a solution to* UE-PE-$\mathcal{V}$. *Then $Z(\boldsymbol{v}^\kappa) \leq (1+\kappa)^{n+1} Z(\boldsymbol{v}^0)$, where $\boldsymbol{v}^0$ is the PRUE flow.*

For general cases, comparing equilibrium flows in $\mathcal{F}$ with those in $\mathcal{F}_{1+\kappa}$, we show that $\sup_\rho \mathsf{PoSat}(\rho) = (1+\kappa)^{n+1}$ under some mild conditions.

Theorem 3.2 *Suppose the link travel time functions are in the polynomial form of $t_a(v_a) = \sum_{m=0}^n b_{am}(v_a)^m$, where $b_{am} \geq 0$. Let $\boldsymbol{f}^\kappa \in \mathcal{F}$ be any κ-SatUE and $\widehat{\boldsymbol{f}}^0 \in \mathcal{F}_{1+\kappa}$ be the PRUE flow. Suppose that $\kappa \geq 0$ is sufficiently small, in particular, so that $\sum_{p\in\mathcal{P}}[c_p(\widehat{\boldsymbol{f}}^0) - c_p(\boldsymbol{f}^\kappa)](\widehat{f}_p^0 - f_p^\kappa) \geq \kappa \sum_{p\in\mathcal{P}} c_p(\boldsymbol{f}^\kappa)\left|\widehat{f}_p^0 - f_p^\kappa\right|$. Then we have $C(\boldsymbol{f}^\kappa) \leq (1+\kappa)^{n+1} C(\boldsymbol{f}^0)$, and $\sup_\rho \mathsf{PoSat}(\rho) = (1+\kappa)^{n+1}$.*

4 Numerical Bound for PoSat

Using UE-PE-$\mathcal{V}$, we formulate the SatUE worst-case total travel time as follows:

$$\max_{\boldsymbol{v}^\kappa, \boldsymbol{\varepsilon}} \quad Z(\boldsymbol{v}^\kappa) = \sum_{a \in \mathcal{A}} t_a(\boldsymbol{v}^\kappa) v_a^\kappa \tag{5}$$

$$\text{subject to} \quad \sum_{a \in \mathcal{A}} (t_a(\boldsymbol{v}^\kappa) - \varepsilon_a)(v_a - v_a^\kappa) \geq 0 \qquad \forall \boldsymbol{v} \in \mathcal{V} \tag{6}$$

$$0 \leq \varepsilon_a \leq \frac{\kappa}{1+\kappa} t_a(\boldsymbol{v}^\kappa) \qquad \forall a \in \mathcal{A} \tag{7}$$

We utilize sensitivity analysis of variational inequalities [4] to develop an algorithm for this problem. We use the Sioux Falls network to compare the SatUE worst-case total travel time with the PRUE total travel time both numerically and theoretically. As Figure 1a shows, as κ increases, the worst-case SatUE total travel time increases, while the PRUE total travel time is constant. As Figure 1b represents, there is a considerable difference between the numerical and analytical bound for worst-case SatUE total travel time.

References

[1] Di, X. and Liu, H. X. (2016). Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Research Part B: Methodological*, 85:142–179.

[2] Lou, Y., Yin, Y., and Lawphongpanich, S. (2010). Robust congestion pricing under boundedly rational user equilibrium. *Transportation Research Part B: Methodological*, 44(1):15–28.

[3] Mahmassani, H. S. and Chang, G.-L. (1987). On boundedly rational user equilibrium in transportation systems. *Transportation Science*, 21(2):89–99.

[4] Patriksson, M. (2004). Sensitivity analysis of traffic equilibria. *Transportation Science*, 38(3):258–281.

[5] Roughgarden, T. and Tardos, É. (2002). How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259.

[6] Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics*, pages 99–118.

[7] Zhu, S. and Levinson, D. (2015). Do people use the shortest path? an empirical test of wardrop's first principle. *PloS One*, 10(8):e0134322.

Time-Dependent Vehicle Routing Problem with Time Windows on a Road Network

Michel Gendreau
Département de mathématiques et de génie industriel and CIRRELT
Polytechnique Montréal

Andrea Lodi
Département de mathématiques et de génie industriel and CIRRELT
Polytechnique Montréal

Jean-Yves Potvin
Département d'informatique et de recherche opérationnelle and CIRRELT
Université de Montréal

Maha Gmira
Département de mathématiques et de génie industriel and CIRRELT
Polytechnique Montréal, Édouard-Montpetit Blvd., Montréal, Canada
Email: maha.gmira@polymtl.ca

1 Introduction

Most vehicle routing problems consider a customer-based graph, where an arc between a pair of customers corresponds to a shortest path in the underlying road network. In a time-dependent context, though, the shortest path is not fixed and may change depending on the time of the day. To account for this reality, we consider the Time-Dependent Vehicle Routing Problem with Time Windows on a Road Network ($TDVRPTW_{RN}$). This problem is defined on a graph $G = (V, A)$, where V is the set of nodes (e.g, street intersections) and A is the set of arcs (e.g., street segments between two intersections). A subset C of nodes corresponds to customers. Each customer $i \in C$ has a demand q_i, a time window $tw_i = [a_i, b_i]$ and a service or dwell time s_i. It should be noted that a vehicle cannot arrive at customer i after the upper bound b_i of the time window, but can arrive before the lower bound a_i, in which case the vehicle has to wait until time a_i to start its service.

Node 0 is the depot where a fleet of vehicles is located, each vehicle having capacity Q. The time window at the depot is $[a_0, b_0]$ where a_0 and b_0 define the beginning and end of the operations day, respectively. A time-dependent speed function is associated with each arc by dividing the day into time slots and by associating a speed with each time slot [1]. The problem is then to generate a set of feasible vehicle routes, starting and ending at the depot, to serve all customers at minimum cost. The latter is the sum of route durations (travel time + waiting time + service time).

We propose a tabu search heuristic to solve the $TDVRPTW_{RN}$. An important contribution of this work is the development of techniques that allow a constant time evaluation of every solution in the neighborhood of the current solution (i.e., without any need to propagate the impact of a modification along a route). This is described in the following.

2 Dominant shortest-path structure

First, a pool of different shortest paths between each pair of customers i and j is generated by considering different starting times during the day and by calculating a time-dependent shortest path for each starting time. Dijkstra's algorithm is used for this purpose, where the travel time extension from one node in the road network to the next along an arc is calculated with the procedure in [1]. For each path in this pool, we generate a piecewise linear function that gives the arrival time at customer j given the departure time at customer i. Then, the dominant shortest path structure is obtained by identifying crosspoints between the generated paths. In Fig. 1, the resulting structure over three different paths between i and j is indicated by a continuous line.

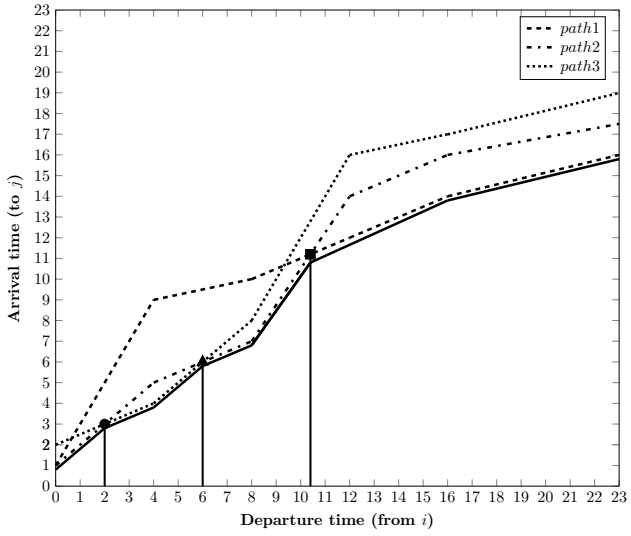


Figure 1 – Dominant shortest path structure between customers i and j

This structure indicates the path to follow to go from customer i to customer j at a given time of the day. It is used to determine the feasibility of a solution in the neighborhood of the current

solution in constant time during the tabu search heuristic. It is used similarly at each iteration of the greedy construction heuristic when an initial solution is created for the tabu search.

3 Problem-solving

3.1 Initial solution

First, a greedy insertion heuristic, where the routes are constructed sequentially, is used to create an initial solution for the tabu search. At each iteration, a customer is randomly selected and its best feasible insertion place in the current route is determined, while checking the feasibility and evaluating the approximate cost of each insertion in constant time (see how it is done in the context of the tabu search in the following section). This is repeated until all customers are visited.

3.2 Tabu search

The initial solution is improved by the tabu search. We focus here on its two most important components: the neighborhood used and the constant time evaluation of each neighbor solution.

3.2.1 Neighborhood structure

The tabu search exploits a neighborhood structure based on CROSS exchanges [3]. This type of exchange has proven to be well suited for problems with time windows because it does not reverse segments of routes. An example of a CROSS exchange is shown in Fig. 2, where the sequences of customers from e to f in Route 1 and from h to k in Route 2 are exchanged.

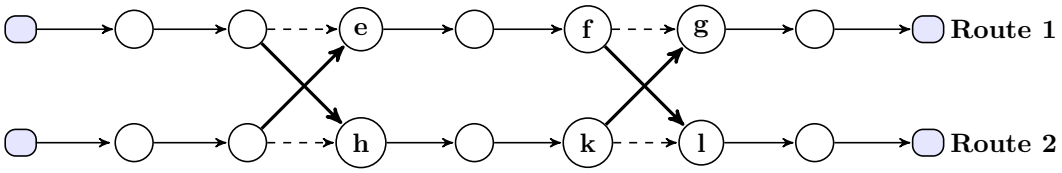


Figure 2 – CROSS exchange

3.2.2 Solution evaluation

When a CROSS exchange is performed, the feasibility of the new solution is checked and its approximate cost is calculated. Both can be done in constant time as it is explained below.

Feasibility

To determine if a neighbor solution is feasible (apart from the capacity constraint which is easily checked), we maintain at each customer i in the current solution the latest departure time $\bar{t}_i$ that allows the route of customer i to remain feasible. This is done while taking into account time-dependency. In particular, the dominant shortest path structure (see Fig. 1) is used to associate

the appropriate shortest path between consecutive customers i and j in a route that matches $\bar{t}_i$ and $\bar{t}_j$. The latest departure time values are then used to evaluate in constant time the feasibility of a neighbor solution. For example, in the CROSS exchange of Fig. 2, the neighbor solution is not feasible if the new departure times at l (now from f) and g (now from k) exceed $\bar{t}_l$ and $\bar{t}_g$, respectively.

Approximate cost

If the neighborhood solution is feasible, its approximate cost must then be evaluated. For this purpose, we maintain at each customer i in the current solution a penalty p_i. This penalty corresponds to the delay incurred at the successor of customer i, if the actual departure time at i is delayed by one time unit. In the CROSS exchange of Fig. 2, if the departure times at customers l (now from f) and g (now from k) are delayed by Δ_l and Δ_g time units, respectively, then the impact on the route cost is estimated at $\Delta_l \times p_l + \Delta_g \times p_g$. The best m feasible neighbor solutions, based on this approximate cost, are then considered and the best one, using the exact cost obtained through propagation along the modified routes, is selected at the end.

4 Experiments

Our tabu search heuristic will be tested on two different sets of test instances. The first set is generated using the procedure described in [2]. There are instances for a road network of 50 nodes with 16 and 23 customers; 100 nodes with 25, 33 and 50 customers; 200 nodes with 25 and 50 customers. The second set of test instances, used in [4], is based on a real network of the central urban area of *Aix-en-Provence*. The network contains $5,437$ nodes and $10,181$ arcs with 5, 10 and 25 customers. Computational results on these two sets of test instances will be presented at the conference.

References

[1] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Vehicle dispatching with time-dependent travel times", *European Journal of Operational Research* 144(2), 379-396 (2003).

[2] A.N. Letchford, S.D. Nasiri and A. Oukil, "Pricing routines for vehicle routing with time windows on road networks", *Computers & Operations Research* 51, 331-337 (2014).

[3] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows", *Transportation Science* 31(2), 170-186 (1997).

[4] H.B. Ticha, "Vehicle Routing Problems with Road-Network Information", Doctoral dissertation, Université Clermont Auvergne, (2017). Retrieved from *https://tel.archives-ouvertes.fr*

Modeling the Operation Dynamics of Ride-sourcing Markets

Xinwu Qian
Lyles School of Civil Engineering
Purdue University, USA

Satish V. Ukkusuri
Lyles School of Civil Engineering
Purdue University, USA
Email: sukkusur@purdue.edu

Rui Chen
Department of Industrial Engineering
Tsinghua University, China

Chao Yang
School of Transportation Engineering
Tongji University, China

1 Introduction

The introduction of various Ride-sourcing (RS) platforms has significantly altered the urban transportation landscape where the taxi market was a monopoly to what is arguably now an oligopoly. To date, we have observed that the substantial differences between RS and traditional taxi service led to disruptive yet controversial changes in the market, but the underlying mechanisms on how these differences drive the dynamics of the market remain unknown. Modeling the operation dynamics for RS is a challenging research problem primarily due to its two distinct operation features: 1) the entry-deregulation scheme and 2) the spatial-temporal surge pricing scheme. Entry-deregulation implies that drivers may join or leave the market at any time, which suggests that the market supply is a variable and depends on the revenue level of the market. However, almost all existing studies consider the supply as a fixed input which is used to constraint the total labor hours in the market [1, 2, 3, 4]. On the other hand, surge pricing introduces the third player (the RS platform) into the market besides the passengers and the drivers. In addition, the spatiotemporally varying price also brings the issue of curse of dimensionality for the modeling of market dynamics. And studies exclusively assume that the price will only vary with time [4, 5, 3], while the whole study area will share the same price multiplier. These simplifications are apparently different from the actual dynamics of RS market following our previous discussion.

The study presents the mathematical model for the daily operations of the RS market, where the market is entry-deregulated and has spatiotemporal dynamic pricing. The competition among the RS operators (the platform), the riders, and the drivers is explicitly considered: the operators propose the spatiotemporal dynamic pricing and subsidy policies to maximize their revenue, riders decide whether or not make the trip based on the perceived travel cost, and the drivers decide whether they will leave/enter the market and the location to pick up riders based on their perceived utility. The game among the three major players is formulated as a dynamic programming problem with equilibrium constraints (DPEC).

We introduce the time-expanded service network to represent such competition at the network level, and propose a rolling-horizon approach for the DPEC by solving a sequence of mathematical problems with equilibrium constraints. We collect real-world cruising data from one RS provider in New York City (NYC) and the data from NYC taxicabs to validate behavioral parameters, calibrate model coefficients, and conduct numerical experiments. Our results suggest that RS markets are self-regulated even though the fleet size and entry are deregulated, and this characteristic contributes to a more efficient market than the traditional taxi industry. We observe that dynamic pricing plays an important role in ensuring the efficient of the system. Besides dynamic pricing, our results also suggest the necessity for subsidizing drivers, which is found to be a win-win strategy for the operators, the passengers, and the riders of the RS market.

2 Method

2.1 Preliminary

We introduce the RS triangle among the passengers, the drivers, and the RS operators (Figure 1). In

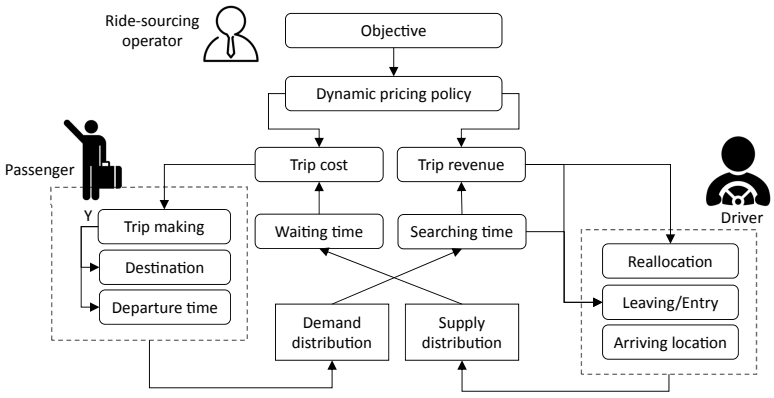


Figure 1: RS Triangle: the decision making framework among the operator, riders, and drivers in the RS market

general, the RS is a two-sided market that is entry-deregulated. It has a fixed time-distance-based trip fare structure, and the price will be dynamically adjusted with spatial-temporally varying price multipliers (PM). The RS operator (the platform), the drivers, and the passengers are seeking to maximize their own utility, which in return affect each others' decision-making behaviors. The objective of the passengers in RS market is to decide whether to make the trip or not, which is primarily affected by the cost of travel and the cost of waiting. The set of decisions made by potential passengers gives rise to the demand distribution, which largely determines the distribution of existing drivers in the market and may also affect the potential entry locations of drivers in the immediate future. The drivers, based on their schedules, tend to play the strategies to maximize their trip revenue. Their strategies may involve a series of decisions, including the time to enter or leave the market, and the locations to receive the orders from the platform. Once the order is dispatched, the drivers should take the passengers to their designated

destination. Upon the completion of assigned trips, the drivers become available and need to determine the next location for receiving future orders, which combined with newly entered and left drivers shape the supply distribution of the RS market. Note that the decision making of passengers and drivers are coupled together since the passengers' waiting time is a function of the number of passengers who request a ride and the availability of nearby drivers, and the drivers' searching cost is a function of the number of drivers that are competing for passengers as well as the availability of potential passengers. On top of the game between passengers and drivers is the decision making process of the RS operator, who designs PMs at different locations and time intervals to achieve their objective (e.g., maximizing total revenue or minimizing empty trips). And the resulting PMs will further impact the trip cost of passengers and trip revenue of drivers, which in return will change the demand and supply distributions of the RS market.

2.2 Modeling approach

It is intuitive that the dynamic pricing problem for RS market can be modeled as a dynamic programming problem. Based on the RS triangle and the assumptions, we presents the general formulation for the RS market with surge pricing as follows:

$$\begin{aligned} \textbf{maximize} \sum_{t=0}^{T} \sum_{i=1}^{N} f_t(P_i^t, D_i^t, \pi_i^t) \\ \text{subject to} \\ \bar{D}_i^t = Q(D_i^t, \pi_i^t), \quad \forall t, i \\ P_i^t = \mathcal{G}(\bar{D}_i^t, P_*^{<t}, P_{-i}^t, \pi_i^t), \quad \forall t, i \end{aligned} \tag{1}$$

Problem 1 seeks to maximize the total system revenue in N zones over T time intervals, and the set of constraints describes the operation dynamics of RS market. In particular, $Q(D_i^t, \pi_i^t)$ represents the induced demand function that models passengers' reactions to the proposed PM π_i^t. As the most crucial component of the dynamic programming problem, $\mathcal{G}(\bar{D}_i^t, P_*^{<t}, P_{-i}^t, \pi_i^t)$ refers to the state transition function, which maps the supply distribution in previous time intervals ($P_*^{<t}$) and present induced demand and PM to the supply distribution of current time interval.

Solving problem 1 is extremely difficult due to the state transition function being an equilibrium constraint. The equilibrium constraint comes from the game among the market operator, supply, and demand as illustrated by the RS triangle. Due to page limit, we only highlight the solution methods developed to solve this problem:

1. A time-dependent service network (TSN) is proposed to characterize the game among the RS players at network level. The network structure is able to capture the entry and leave behavior of drivers and the implementation of spatial-temporal varying PM.

2. We decompose the dynamic programming problem with equilibrium constraints into solving a series of mathematical programming problem with equilibrium constraints (sub-horizon game), based on the developed TSN and the rolling-horizon heuristic.

3. We introduce novel functions to capture the leaving and entry behavior and the searching behavior of drivers. These functions are calibrated using real-world RS driver data.

4. We prove the solution existence of the problem and develop an algorithm to find the strongly stationary point for each sub-horizon game.

3 Results

We introduce two networks to validate the quality of our solution algorithm and demonstrate the value of our model. The first network is a four-nodes toy network, where we obtained the following major findings:

1. The algorithm scales very well with increasing size of sub-horizon games and each sub-horizon game can be solved efficiently.
2. The TSN-based rolling-horizon algorithm may find near optimal solution as compared to solving the original problem directly.

The second network is an abstraction of NYC with 81 OD pairs. We use real-world trip data as input. The key findings are summarized below:

1. Though RS market is deregulated, the RS supply is self-regulating and adaptive to the demand distribution. The operation efficiency of RS market is therefore much better than traditional taxi industry with less significant over-supply and under-supply issues.
2. Dynamic pricing is essential to RS market for effective supply management and revenue optimization. Dynamic pricing helps to induce additional drivers during supply shortage, and effectively restricts greedy behavior of drivers when demand and supply are aligned.
3. We find that subsidizing drivers will be an effective complement to the dynamic pricing strategy. Introducing subsidy for drivers will improve total revenue for both operators and drivers, contribute to serving more passengers, and reduce the passengers' out-of-pocket cost.

References

[1] H. Yang, S. C. Wong, A network model of urban taxi services, Transportation Research Part B: Methodological 32 (4) (1998) 235–246.

[2] F. He, Z.-J. M. Shen, Modeling taxi services with smartphone-based e-hailing applications, Transportation Research Part C: Emerging Technologies 58 (2015) 93–106.

[3] X. Qian, S. V. Ukkusuri, Taxi market equilibrium with third-party hailing service, Transportation Research Part B: Methodological 100 (2017) 43–63.

[4] L. Zha, Y. Yin, Y. Du, Surge pricing and labor supply in the ride-sourcing market, Transportation Research Part B: Methodologicaldoi:https://doi.org/10.1016/j.trb.2017.09.010.
URL http://www.sciencedirect.com/science/article/pii/S0191261517307683

[5] X. Qian, S. V. Ukkusuri, Time-of-day pricing in taxi markets, IEEE Transactions on Intelligent Transportation Systems 18 (6) (2017) 1610–1622.

Traffic-dependent limited unfairness in a system optimum traffic assignment

E. Angelelli **V. Morandi**

M.G. Speranza

Department of Economics and Management, University of Brescia

Email: grazia.speranza@unibs.it

1 Introduction

Urban areas are world-wide affected by severe road congestion problems. Traffic congestion is a result of an economic growth that causes an increase of private and commercial transportation. Almost all vehicles are nowadays equipped with sat-nav devices that can also display the current traffic flows and return the fastest path for the vehicle. The route suggested by these devices does not consider the impact of simultaneous individual choices on the congestion of the road network. For example, all commuters entering the network in the same point and heading to the same destination have the same information and, consequently, the same path will be suggested to them generating congestion, even though multiple paths with similar travel time may be available. Coordination among vehicles is a potentially powerful tool to prevent congestion.

Traditionally, traffic assignment concerns assigning routes to users (vehicles with, and in the future possibly without, drivers) and is usually defined on a road network with an origin-destination (OD) matrix specifying the demand for transportation, i.e, the number of vehicles per time unit that is expected to travel from each origin to each destination. The objective is to minimize the total travel time experienced by users in the system. Experienced travel times are computed for each network arc a through the so-called *latency function* $t_a(x)$, which depends on the arc traffic flow, or simply flow, x.

Traffic assignment models were first presented in the seminal work [4] where the two most famous principles on traffic assignment (the user equilibrium and the system optimum) are stated. The user equilibrium represents an assignment in which the travel times along all used routes from an origin to a destination are equal and not more than the travel time that would be experienced by a single user on any other route. On the other hand, the system optimum is an assignment in which the total travel time is minimized. The difference in terms of total travel time between

implementing a user equilibrium and a system optimum traffic assignment is called price of anarchy and is well-known in the literature. While the user equilibrium ensures fairness for users travelling between the same origin and destination, in a system optimum traffic assignment some users may be assigned to paths that require much more time than paths assigned to other users for the same OD pair, generating a high level of unfairness among users.

In order to reduce the total travel time while maintaining fairness among users, a constrained system optimum traffic assignment model was first presented in [3]. The model is convex non-linear and considers as eligible those paths that have a normal length within a certain percentage of the path with the shortest normal length. The normal length of an arc (and thus of a path) is an a priori estimate of its travel time. The authors propose different options to compute this measure such as the Euclidean length, the free-flow travel time and the travel time under user equilibrium. The first attempt to use a linear programming model to solve the constrained system optimum traffic assignment problem is presented in [1]. The total travel time is minimized while keeping the network non-congested, if possible, or at its minimum congestion level, otherwise. The set of eligible paths is restricted as in previous works. In [2] a linear programming model, in which a traffic-dependent latency function is embedded, is presented. The proposed model adopts a piecewise linear approximation of the convex latency function that makes use of continuous variables only.

All these models assign paths to OD pairs identifying a priori the set of eligible paths. However, when traffic flows on the road network, a path that is eligible a priori could turn out to be more unfair than expected a priori. Moreover, there may be paths not considered a priori that, if considered, would be fair. To the best of our knowledge, no model for the traffic assignment problem has been proposed that limits the unfairness experienced a posteriori by the users.

In this paper, a model is proposed that embeds in the mathematical programming formulation constraints ensuring that the traveling time of used paths does not exceed by a given percentage the fastest path from origin to destination. The resulting model requires the introduction of binary variables. The quality of a path for a user traveling from an origin to a destination is measured through the so called *fastest path unfairness*, that is the relative difference between its traveling time and the traveling time on the fastest path, computed on all paths, used and unused. The traveling times are, in all cases, traffic-dependent. A mixed-integer linear programming (MILP) model, called *Fastest Path Unfairness Constrained System Optimum* (FP-UC-SO) model, is presented that minimizes the total travel time experienced by the users while bounding the fastest path unfairness of each used path to be lower than a maximum threshold called *maximum fastest path unfairness*. A variant of the FP-UC-SO model, called *Loaded Unfairness Constrained System Optimum* (L-UC-SO), is also presented which uses the *loaded unfairness* measure for a path which compares the traveling time on a path with the fastest path actually used by some users

(see [3]). The FP-UC-SO and L-UC-SO models require the a priori enumeration of all possible paths from origin to destination. Each path is associated with a binary variable. A piecewise linear approximation of the traffic-dependent latency function ensures the linearity of the model. A matheuristic is proposed for the solution of the models and computational results are obtained on benchmark instances.

2 The models and the matheuristic

Let $G = (V, A)$ be a directed network, where V and $A \subseteq V \times V$ represent, respectively, the set of vertices and the set of arcs. Arcs represent road segments while vertices represent junctions between roads and/or an origin or destination point for an OD pair. A latency function $t_{ij}(x_{ij})$, representing the arc travel time depending on the rate of vehicles x_{ij} entering the arc, is consistently associated with each arc $(i, j) \in A$. In addition, each arc is associated with a number of parameters as the free-flow travel time t_{ij}^{FF} $(= t_{ij}(0))$ and a tuning parameter u_{ij} used to shape the latency function. The most popular latency function is the U.S. Bureau of Public Road (BPR) function $t_{ij}(x_{ij}) = t_{ij}^{FF}[1 + 0.15(\frac{x_{ij}}{u_{ij}})^4]$ and is the one used in the model. Transportation demand rates are represented by the set C of origin-destination (OD) pairs. Each OD pair $c \in C$ is associated with an origin $O_c \in V$, a destination $D_c \in V$, and a demand rate d_c from O_c to D_c. The set of paths from O_c to D_c is denoted by K_c. An indicator a_{ij}^{kc} takes value 1 if path $k \in K_c$ contains arc $(i, j) \in A$ and takes value 0, otherwise. The *maximum fastest path unfairness* allowed is denoted by ϕ.

The total arc travel time $t_{ij}(x_{ij})x_{ij}$ is linearized as follows. An upper bound U_{ij} on the flow rate x_{ij} is fixed and each non-linear term $t_{ij}(x_{ij})x_{ij}$ is linearized on the range $[0, U_{ij}]$ by a piecewise linear function. The FP-UC-SO model turns out to be a MILP model, where a binary variable is associated with each path.

If the *loaded unfairness* is used to measure the fairness of a path, the relative difference between its traveling time and the traveling time on the fastest path is computed considering only the actually used paths. In the L-UC-SO model, the *maximum loaded unfairness* is denoted by ϕ. Also the L-UC-SO model turns out to be a MILP model.

The FP-UC-SO and L-UC-SO models contain a huge number of variables and constraints for each OD pair c. For this reason, a heuristic, called *Path Construction Matheuristic* (PC-M) algorithm, has been developed.

We refer here to the FP-UC-SO model only. Let Rel-UC-SO denote the linear relaxation of the FP-UC-SO model. The PC-M algorithm starts considering only the fastest paths under free-flow condition for each OD pair, solves the Rel-UC-SO model restricted to these paths and uses the values of the variables to weight an auxiliary network on which a path search is performed.

A new set of paths is identified and the procedure is repeated until no new paths can be found. Eventually, a restricted version of the FP-UC-SO model is solved.

3 The results

All models have been solved using CPLEX 12.6.0 on a Windows 64-bit computer with Intel Xeon processor E5-1650, 3.50 GHz, and 64 GB RAM. To speed up the solution of the PC-M algorithm, a relative gap tolerance on the MILP solver has been set to 1%, which was shown through preliminary experiments to achieve a good trade-off between speed and solution quality. For all the experiments, the BPR latency function has been used with $U_{ij} = 4u_{ij}$.

A set of 32 instances with 24 nodes, different in terms of geography and demand pattern, 4 increasing size instances with 270, 300, 330 and 360 nodes and 2 real-world instances have been used in a computational study to assess the performance of the FP-UC-SO and the L-UC-SO models, and of the PC-M algorithm.

The computational results show that the models are effective in terms of trade-off between travel time and fairness. Moreover, the average and maximum errors produced by the heuristic, with respect to the optimum, are 0.22% and 1.55%, respectively.

References

[1] E. Angelelli, I. Arsik, V. Morandi, M. Savelsbergh, M.G. Speranza, "Proactive route guidance to avoid congestion", Transportation Research Part B: Methodological, 94, 1-21, 2016.

[2] E. Angelelli, V. Morandi, M. Savelsbergh, M.G. Speranza, "System optimal routing of traffic flows with user constraints using linear programming", under review.

[3] O. Jahn, R.H. Möhring, A.S. Schulz, N.E. Stier-Moses, "System-optimal routing of traffic flows with user constraints in networks with congestion", Operations Research, 53, 600-616, 2005.

[4] J.G. Wardrop, "Road paper. Some theoretical aspects of road traffic research", Proceedings of the Institution of Civil Engineers, 1, 325-362, 1952.

A mixed integer programming approach for scheduling spatially distributed jobs with degradation rate: application to pothole repair

Rajan Batta and Fatemeh Aarabi
Department of Industrial and Systems Engineering
University at Buffalo
Buffalo, New York 14051
USA
Email: batta@buffalo.edu

1 Introduction

In classical job scheduling, the processing time of jobs is considered constant. However, in many real-world applications processing time is an increasing function of when the job is started. Fire-fighting and medical emergencies are instances of jobs where delayed response leads to a larger amount of processing time (Wang et al. 2018). The application that motivated our work is pothole repair. It is a classic case of scheduling deteriorating jobs since the chemicals that bind pavement deteriorate over time making pothole repair an increasing function of time. Providing humanitarian relief to victims after a disaster also belongs to this class of scheduling problems, since more time for treatment is needed if medical attention is delayed. In these situations, the actual processing time is larger than the normal service time and is a function of degradation rate and start time (Kunnathur and Gupta, 1990).

Another set of applications of scheduling problems are jobs that are distributed spatially. In many machine scheduling problems, the location of machines are considered fixed and jobs move through them, but there are many occasions that jobs are fixed and machines are mobile. Emergency response is an example where jobs are distributed over a spatial region and mobile servers serve them. Other examples include humanitarian relief supply and pothole repair. All of these examples need to consider the network-based nature of the problem and use solution algorithms that consider distances that servers has to traverse to access jobs. The most basic algorithm that

is useful in solving spatial problems is the traveling salesman problem.

In this paper, we use an MIP formulation to find the optimal schedule for a set of spatially distributed deteriorating jobs. Its objective is to minimize the sum of the processing times of all jobs plus the sum of the travel time for all servers.

Our main contributions are as follows:

1. We present a scheduling problem for spatially distributed deteriorating jobs.
2. We model job deterioration continuously over time such that it occurs even when servers are idle.
3. We demonstrate an application of our model for pothole repair, and explore features that allow consideration of continuously arriving jobs and equity considerations between impacted regions.

2 Formulation

Consider the scheduling of n spatially distributed jobs with specified degradation rates and k servers. Let $G = (N, A)$ be a complete undirected graph with node set $N = \{0, 1, 2, ..., n, n+1\}$ and set of links $A = \{(i, j) \mid i, j \in N, i \neq j\}$. In this graph, node 0 represents the beginning depot for each day and node $n + 1$ represents the ending depot for each day, and other nodes represent jobs. Note that nodes 0 and $n + 1$ do not have any processing time associated with them. We now focus on non-depot nodes. For node i we consider a_i as the fixed time required to process it and α_i as its degradation rate, what we mean by this is that if job i starts processing at time w, its processing time is $a_i + \alpha_i w$ (i. e. linear degradation). For every link $(i, j) \in A$ let τ_{ij} be the travel time between node i and node j. We let k denote the number of identical servers. Let m be the number of days or time periods (each 24 hours duration) over which the jobs must be processed. Each server is available only during the first T hours of each day, with $T \leq 24$. Each day, every server leaves the depot at the beginning of the day, travels to its assigned jobs in the specified sequence, processes each of its assigned jobs, and returns to the depot within T hours.

To formulate the mathematical programming model, we use the following decision variables. Binary variable x_{ijt} is equal to 1 if a server travels from node i to node j during day t, 0 otherwise. Continuous variable w_{it} specifies the time that a server arrives to start the job associated with node i in day t. For space reasons we do not present our formulation. We do note that our formulation does not have an index for servers. This is because its solution provides us with the sequence of jobs for each server for each day. We chose not to have an index for servers as it greatly reduces

computation time for solving.

Intuitively, jobs with high degradation rate should be done sooner, as our model seeks to minimize the sum of job processing times as one component of its objective. Also, a job with less travel time to reach it should be done earlier, as this assures an earlier start to jobs and hence less processing time. So if there is a match between the degradation rate ordering and the travel time ordering we have an optimal solution to our model. This condition is rare but provides a useful basis for our greedy heuristic. In general, the optimal route is some combination of VRP considerations (sum of travel times for servers) as well as scheduling of deteriorating job considerations (sum of job processing times).

3 Computational study

We have done extensive computational testing of the model. This includes development of a chronological decomposition heuristic, a comparison of the greedy versus the chronological decomposition heuristic, and studying the impact of key parameter values on the objective function, for which we performed a factorial design experiment, which studies one-way, two-way and three-way effects. We are interested in finding the impact of three parameters of our model. Therefore, we ran a 3^3 full factorial design. Here l^k represents a factorial design in which there are k factors that each one has l levels. We considered the following three factors for our design: degradation rate (factor A) with three levels of low, medium and high; distance between jobs (factor B) with three levels of short, medium and long; and the number of hours that each worker can work during a day (factor C) which we considered three levels of 8, 12, and 16 hours. Our results show hat degradation rate and work hours have the largest impact.

4 Pothole repair case study

To demonstrate applicability of our model we have studied the case of pothole repair. Pothole repair is usually done after the winter season is over. It consists of several spatially distributed jobs (potholes) at various points in the transportation network under study. These jobs are also degrading because potholes tend to become larger and more time consuming to fix if their repair is delayed. The community our university is housed in, Buffalo, New York, is a classic pothole prone area because it receives an average of 93 inches of snowfall in the winter, and a typical winter day has daytime high temperatures above freezing and night time low temperatures below freezing. These constant melting and freezing cycles rapidly create potholes and their size increases as the

winter progresses, leaving significant work for the road repair crews at the end of the winter season.

We gathered data for our case study from many different sources. Degradation rates were assumed to be a function of the annual average daily traffic on the road segment. Initial repair times were estimated using a queuing model. Equity considerations fro pothole repair were investigated by adding suitable constraints to the model. The case of continuously arriving jobs was also investigated. The impacts of degradation, spatial distribution and equity were isolated so as to better understand their individual impacts.

5 Conclusions and ongoing work

Our conclusions are as follows: (i) It is important to consider the elements of spatial distribution of jobs and degradation of jobs together in one integrated model. (ii) It is important to incorporate equity considerations.(iii) It is important to address situations where jobs arrive continuously over time.

Several extensions are being investigated. The entire problem can be viewed as a queue control problem when jobs arrive continuously over time and theories from spatially distributed queues could potentially be applied. We are presently working on this aspect. We have uncovered some interesting managerial insights in this regard. Server cooperation is viewed to be a highly positive aspect of spatial queues, as it improves overall performance. With degradation of a job occurring, server cooperation has to be carefully done, because placing a server in a poor location can lead to large travel times and hence significant job degradation. This job degradation in turn leads to further server unavailability, a domino effect. One way to combat this is to refer calls to an outside service for a penalty cost in certain situations as opposed to insisting on server cooperation to respond to calls when a server is available.

The model can be specialized for many different application settings, such as forest-fire control or emergency response. We have plans to work on this aspect in the near future.

References

T. Wang, R. Baldacci, A. Lim, and Q. Hu. A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine. European Journal of Operational Research, 2018.

A. S. Kunnathur and S. K. Gupta. Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. European Journal of Operational Research, 47(1):5664, 1990.

Matching Passengers and Drivers with Multiple Objectives in Ride Sharing Markets

Guodong Lyu, Chung Piaw Teo
Department of Analytics and Operations
National University of Singapore

Wangchi Cheung
Department of Industrial Systems Engineering and Management
National University of Singapore

Hai Wang
School of Information Systems
Singapore Management University
Email: haiwang@smu.edu.sg

In many cities in the world, ride sharing companies, such as Uber, Didi, Grab and Lyft, have been able to leverage on Internet-based platforms to conduct online decision making to connect passengers and drivers. These online platforms facilitate the integration of passengers and drivers' mobility data on smart phones in real-time, which enables a convenient matching between demand and supply in real time. These clear operational advantages have motivated many similar shared service business models in the public transportation arena, and have been a disruptive force to the traditional taxi industry.

Matching passengers (demand) with drivers (supply) in real-time is a challenging problem for the ride sharing platforms. Greedy policy, as a common used benchmark matching policy, assign passenger to the "nearest" available driver, based on the pick-up time estimated from each driver's location and surrounding traffic conditions. Note that the pick-up time of the assigned driver affects whether a passenger will cancel the booking, or show up at the pick up location. In reality, platforms also consider many other objectives in the matching policy. One important objective is the rating for drivers. Uber, for instance, uses rating provided by passengers to rate the drivers, and booted those drivers whose rating fall below a threshold from their system. Platforms usually give priority to drivers with higher ratings in the matching policy, especially during the off-peak

hours with sufficient supply. Such preference to drivers with higher rating could encourage drivers to provide better service, which will also improve the overal service quality in the system. Another important objective is the passenger revenue (i.e., the order estiamted fare). Platforms also give priority to passengers with higher revenue in the matching policy, especially during the peak hours with a large number of passengers. Such preference to passengers with higher revenue could bring more revenue and profit to the platforms.

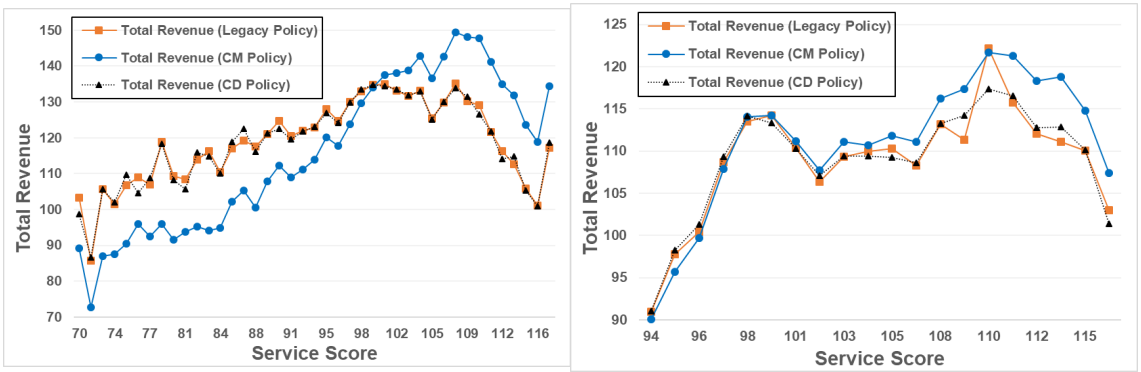
In general, the matching decisions between drivers and passengers are supposed to take the trade-offs between multiple objectives into account. Although piles of efforts have been devoted to designing matching policies for the two-sided sharing market, the majority of these works focused on a single-objective optimization problem. For example, Zhang et al. (2017) [1] develop a batch matching system, with the objective to maximize the driver acceptance rate for each order. Different from the traditional one-order-to-one-driver matching mechanism, they dispatch each order to multiple drivers and let drivers compete for the order. Hu and Zhou (2016) [2] study the dynamic matching control of a two-sided, discrete-time matching system in which both the supply and demand may leave the platform if the wait time before getting passengers or drivers are too long, with the objective to maximize the expected total discounted profit. Ozkan and Ward (2016) [3] propose a linear programming based matching policy that accounts for temporal changing demand and supply and customer patience, with the objective of maximizing the overall number of passengers being served. Wang et al. (2017) [4] introduce the concept of stability in dynamic ride sharing and provide mathematical programming approaches to solve stable and nearly stable ride-share matching problems, with the objective of minimizing the pick-up detour distance. However, few studies shed light on the multi-objective matching policy in the ride sharing markets.

In this paper, we study the matching problem for ride-sharing platform with multiple objectives, and design an online matching policy that simultaneously achieves multiple objectives in a balanced manner. More precisely, we aim to achieve a solution that has the smallest deviation, based on some pre-determined distance function, to an "utopia point", i.e., an ideal solution maximizing the performance of all objectives, but is otherwise non-attainable at the same time. The obtained solution with shortest deviation to the target is called the "compromise solution". We apply the online policy in ride sharing market settings, and provide an online matching policy that simultaneously incorporates driver service scores (driver ratings), pick-up distances and passenger revenues. To be more specific, the platforms would want to dispatch more passenger orders to drivers with higher service rating. This helps to retain the better drivers in the system, and provide better service experience to the customers. However, this could not come without sacrificing the average pick-up distance between dispatched drivers and passengers. Moreover, the platform needs to manage the impact on the bottom line - longer waits lead to lower answer count (passengers drop the bookings) and lower revenue. To balance these different Key Performance Indexes (KPIs),

three key considerations need to be taken into account to design the matching policies in these markets: (1) Passengers with higher revenues should be served with higher priority; (2) Passengers' waiting time for pick-up should be as small as possible; (3) Drivers with higher scores should be dispatched with higher priority.

Note that the traditional approach to multi-objective optimization problem entails a delicate selection of weighting function to aggregate the multiple objectives into a single one, and the central issue there is the choice of the weighting function to be used for aggregation. Our approach exploits the multiple period setting, and the existence of natural performance targets (i.e., the utopia point), to develop an adaptive weighting function that learns from historical performance to drive the algorithm towards the compromise solution. Our detailed numerical studies on the driver dispatching problem show that this approach is able to learn from data the appropriate weighting function that can be used in each period to guide the system towards a good matching solution.

We extracted real world data from Didi Chuxing, the largest on-demand ride sharing platform in China, and conducted industrial implementation of the proposed matching policy. Compared to legacy policies currently in use, such as the weighted average policy (Legacy Policy) or the "closest distance" policy (CD Policy), we observe that all parties in the ride-sharing eco-system, from drivers, passengers, to the platform, are better off under our proposed online matching policy generating the compromise solution (CM Policy): (1) drivers with higher service scores are dispatched with more orders; (2) passengers are more likely to be matched to drivers with higher service scores, and passengers with higher revenues (longer travel distances) are served with higher answer rates; (3) the platform obtains a higher revenue and better long-term brand reputation. For instance, we observe that more jobs are assigned to drivers with higher service quality. Figure 1 demonstrates that expected total revenue earned by drivers with higher service scores (e.g., higher than 101) increases under the CM policy. We also find that the revenue increment for these drivers is indeed due to more orders being dispatched to them. This outcome would motivate drivers to increase their service score by providing better ride sharing service to passengers. We observe a decreasing trend in total revenue for these drivers with extreme high service scores. One possible explanation is that a large proportion of drivers are part-time and their revenue also depends on their total business hours (i.e., active time as a driver on the platform). The dataset reveals this pattern: these drivers with service scores in the interval [98, 108] are more active than the ones with scores in the interval [109,116]. Even so, our proposed policy dispatches more orders to these drivers with higher service scores consistently. As a side effect, the total revenue obtained by the platform during the whole day under the our policy also increases by 0.26% and 0.56% in two cities, respectively.



(a) City A

(b) City B

Figure 1: Driver Service Score vs. Driver Revenue

References

[1] Zhang, Lingyu and Hu, Tao and Min, Yue and Wu, Guobin and Zhang, Junying and Feng, Pengcheng and Gong, Pinghua and Ye, Jieping, "A taxi order dispatch model based on combinatorial optimization", *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2151–2159 (2017).

[2] Hu, Ming and Zhou, Yun, "Dynamic type matching", *Working Paper*, (2016).

[3] Ozkan, Erhun and Ward, Amy, "Dynamic matching for real-time ridesharing", *Working Paper*, (2017).

[4] Wang, Xing and Agatz, Niels and Erera, Alan, "Stable matching for dynamic ride-sharing systems", *Transportation Science*, forthcoming (2017).

Identifying Compliant Users Needed for Social Optimum Routing in Traffic Networks

Tarun Rambha
Department of Civil Engineering
Indian Institute of Science
(Corresponding author email: tarunrambha@iisc.ac.in)

Michael Albert
Darden School of Business
University of Virginia

Guni Sharon
Department of Computer Science and Engineering
Texas A & M University

Stephen D. Boyles
Department of Civil, Architectural, and Environmental Engineering
The University of Texas at Austin

Peter Stone
Department of Computer Science
The University of Texas at Austin

1 Introduction

When self-interested travelers make route choices in a traffic network, a state of Wardrop user equilibrium (UE) is reached in which agents cannot do better by shifting paths unilaterally. It is well known that this state is inefficient compared to a system optimal (SO) flow which minimizes the total travel times of users. At the SO state, all travelers experience equal and minimal marginal costs, but self-interested travelers have an incentive to shift to shorter paths. Congestion pricing can theoretically align self-interested travelers with an SO flow, but this solution is politically unpopular. We thus study the problem when only a subset of travelers can be controlled, perhaps

by providing a monetary incentive for voluntary participation. Finding the optimal flow with a fraction of centrally-controlled agents in a traffic network can be modeled as a Stackelberg game, in which the "compliant" agents are routed by a leader, and the "non-compliant," self-interested agents form a user equilibrium in response ([1]–[3]). In this context, some recent studies focus on identifying the compliant agents for different origin-destination (OD) pairs (see [4] and [5]). In our earlier work on this topic, we addressed this problem of achieving the SO state by minimizing the total number of compliant agents using a linear program (LP) [5]. Results on standard transportation test networks indicated that the percentage of compliant users required increases with network size and varies from 13% (Sioux Falls) to 54% (Chicago Regional).

In this paper, we extend our prior work by further considering the variability in the value of time (VoT) of drivers. Understanding the relationship between VoT and the set of compliant agents necessary to achieve system optimum is critical: to the extent that voluntary participation in the system relies on a monetary incentive, the incentive needed to induce participation likely depends on the traveler's VoT. Different individuals may respond differently to incentives, and there may be a "cheaper" subset of agents to target for compliance. Our proposed model will not only identify the spatial distribution of compliant users needed for an SO state but will also suggest which groups to target. In the following sections, we fist present the idea of finding the fewest compliant travelers in homogeneous settings, and then we provide a sketch of our extensions to the heterogeneous case.

2 Network with Homogeneous Travelers

Consider a network $G = (N, A)$ where N and A are the set of nodes and links respectively. Assume that $Z^2 \subseteq N \times N$ represents the set of OD pairs and the demand between $(r, s) \in Z^2$ is d_{rs}. In this section, we assume that the demand is homogeneous in VoT, and that units are chosen so that VoT is uniformly 1. The travel time on link (i, j) is assumed to depend on its flow x_{ij} as a nonnegative, continuous, increasing function $t_{ij}(x_{ij})$. Under these conditions, it is well known that the UE and SO solutions respectively minimize the convex functions $\sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(x)\ dx$ and $\sum_{(i,j)\in A} x_{ij} t_{ij}(x_{ij})$, and that these solutions exist and are unique in the link flows $\mathbf{x}$.

Denote the SO solution as $\mathbf{x}^*$, and let t^*_{ij} denote the link travel time on (i, j) at the SO solution. These are the link flows and travel times we wish to manifest in the network. Given these SO travel times, let P^{UE}_{rs} represent the set of r-s shortest paths with respect to the link travel times t^*_{ij}, and let P^{SO}_{rs} represents the set of r-s shortest paths with respect to the link marginal costs $t^*_{ij} + x^*_{ij} t'_{ij}(x^*_{ij})$. Under our proposed system, non-compliant agents will select paths from P^{UE}_{rs}, while compliant agents may be assigned to paths belonging to P^{SO}_{rs}.

Finding the least number of compliant drivers is in fact equivalent to selecting an SO path

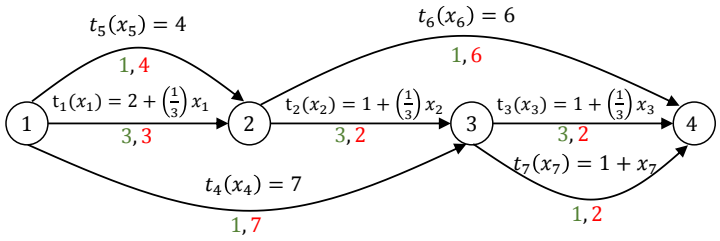


Figure 1: Finding the minimum number of compliant drivers.

Table 1: Three path flow solutions.

Path	1-2-3	1-2-7	5-2-3	5-2-7	4-3	1-6	4-7	5-6
Travel time	7	7	8	8	9	9	9	10
Flow (a)	3	0	0	0	0	0	1	1
Flow (b)	1	1	1	0	1	1	0	0
Flow (c)	2	0	1	0	0	1	1	0

flow solution which maximizes the number of drivers using paths in P_{rs}^{UE}. For such an SO path flow, drivers using paths not in P_{rs}^{UE} experience longer travel times compared to paths in P_{rs}^{UE} and must hence be complaint. Maximizing the number of travelers choosing paths in P_{rs}^{UE} will thus minimize the total number of compliant agents needed for achieving a SO state. To illustrate this idea, consider the following example adapted from Zangui et al. [6]. Suppose there are 5 travelers between 1 and 4 in Figure 1. The delay functions are indicated above the links, and the SO link flows x_{ij}^* and link travel times t_{ij}^* are respectively shown in green and red. There are multiple SO path flows solutions which produce the SO link flows $\mathbf{x}^*$, three of which are shown in Table 1.

The set P_{14}^{UE} is {1-2-3, 1-2-7} and the flow pattern (a) loads maximum travelers on these paths compared to the others. Thus, we need only two compliant drivers in flow pattern (a) versus three in patterns (b) and (c). From a practical perspective, algorithms involving path flows are computationally prohibitive. Instead, we formulate an LP using an origin-based approach to address this issue.

For each origin r, the following formulation maximizes the flow f_{rs} on P_{rs}^{UE} by loading them one origin at a time on a sub-network consisting of links A^r for each origin. This set is constructed by finding links which have zero reduced-cost links with respect to weights t_{ij}^* and $t_{ij}^* + x_{ij}^* t'_{ij}(x_{ij})^*$. (A link (i,j) is said to have zero reduced cost with respect to a vector of link weights $(c_{ij})_{(i,j)\in A}$ if $\mu_i^r + c_{ij} = \mu_j^r$, where μ_i^r represents the length of the shortest path from r to i.) The variables x_{ij}^r represent origin-based compliant link flows and the aggregate link flows are bounded above by the SO solution.

$$\max \sum_{(r,s)\in Z^2} f_{rs} \tag{1}$$

$$\text{s.t.} \sum_{j:(i,j)\in A^r} x_{ij}^r - \sum_{h:(h,i)\in A^r} x_{hi}^r = \begin{cases} \sum_{s\in Z} f_{is} & \text{if } i = r \\ -f_{ri} & \text{if } i = s \\ 0 & \text{otherwise} \end{cases} \quad \forall\, r \in Z \tag{2}$$

$$\sum_{r\in Z} x_{ij}^r \le x_{ij}^* \quad \forall\, (i,j) \in A \tag{3}$$

$$x_{ij}^r \ge 0 \quad \forall\, (i,j) \in A, r \in Z \tag{4}$$

$$0 \le f_{rs} \le d_{rs} \quad \forall\, (r,s) \in Z^2 \tag{5}$$

3 Network with Heterogeneous Travelers

When travelers have different VoT, the system optimum solution minimizes the expected perceived cost $\sum_{(i,j)\in A} \bar{\alpha}_{ij} x_{ij} t_{ij}(x_{ij})$, where $\bar{\alpha}_{ij}$ is the average value of time of users on link (i,j) [7]. The optimum solution to this problem can be computed using Dial's bicriterion traffic assignment model. If the VoT distribution is discrete and finite, we can easily find the demand associated with each VoT value. In this case, denote the support of the VoT is $\{\alpha_1, \ldots, \alpha_{|K|}\}$. Its probability mass function can be used to find the demand of the corresponding class of travelers d_{rs}^k and the earlier LP formulation can be extended to find compliant flows for different VoT classes in the following way. The set A^{rk} denotes the subset of links which have zero reduced cost with respect of weights t_{ij}^* and $\alpha_k t_{ij}^* + \bar{\alpha}_{ij} t'_{ij}(x_{ij}^*) x_{ij}^*$. As before, assume that the SO link flow solution is $\mathbf{x}^*$.

$$\max \sum_{k\in K} \sum_{(r,s)\in Z^2} f_{rs}^k \tag{6}$$

$$\text{s.t.} \sum_{j:(i,j)\in A^{rk}} x_{ij}^{rk} - \sum_{h:(h,i)\in A^{rk}} x_{hi}^{rk} = \begin{cases} \sum_{s\in Z} f_{is}^k & \text{if } i = r \\ -f_{ri}^k & \text{if } i = s \\ 0 & \text{otherwise} \end{cases} \quad \forall\, r \in Z, k \in K \tag{7}$$

$$\sum_{k\in K} \sum_{r\in Z} x_{ij}^{rk} \le x_{ij}^* \quad \forall\, (i,j) \in A \tag{8}$$

$$x_{ij}^{rk} \ge 0 \quad \forall\, (i,j) \in A, r \in Z, k \in K \tag{9}$$

$$0 \le f_{rs}^k \le d_{rs}^k \quad \forall\, (r,s) \in Z^2, k \in K \tag{10}$$

We investigate the results of this formulation for different test networks for an assumed VoT distribution. We also consider a modified objective which minimizes a weighted sum of flows using the VoT values to reflect the level of incentivization. Extensions to problem instances with continuous VoT models will also be presented.

References

[1] P. T. Harker, "Multiple equilibrium behaviors on networks", *Transportation science* 22(1), 39-46 (1988).

[2] T. Van Vuren, D. Van Vliet and M. J. Smith, "Combined equalibrium in a network with partial route guidance", In Traffic control methods. Proceedings of the Fifth NG foundation conference, Santa Barbara, California, USA, 1990.

[3] H. Yang, X. Zhang, and Q. Meng, "Stackelberg games and multiple equilibrium behaviors on networks", *Transportation Research Part B: Methodological* 41(8), 841-861 (2007).

[4] K. Zhang and Y. M. Nie, "Mitigating the impact of selfish routing: An optimal-ratio control scheme (ORCS) inspired by autonomous driving", *Transportation Research Part C: Emerging Technologies* 87, 75-90 (2018).

[5] G. Sharon, M. Albert, T. Rambha, S. Boyles and P.Stone, "Traffic optimization for a mixture of self-interested and compliant agents", In Proceedings of the 32nd conference on Artificial Intelligence (AAAI) New Orleans, Lousiana, USA, February 1990.

[6] M. Zangui, H. Z. Aashtiani, S. Lawphongpanich and Y. Yin, "Path-differentiated pricing in congestion mitigation", *Transportation Research Part B: Methodological* 80, 202-219 (2015).

[7] R. B. Dial, "Network-optimized road pricing: Part I: A parable and a model", *Operations Research* 47(1), 54-64 (1999).

Spatial and temporal synchronization of truck platoons

Anirudh Kishore Bhoopalam, Niels Agatz, Rob Zuidwijk
Rotterdam School of Management, Erasmus University
Corresponding author e-mail: kishorebhoopalam@rsm.nl

1 Introduction

New automated driving technology allows trucks to be virtually linked to drive behind one another in platoons [1]. This means the first truck takes the lead and the following trucks automatically brake, steer, and (de)accelerate based on the leader.

Truck platooning allows for a reduction in the fuel consumed and consequently, the emissions. Studies have shown fuel savings of six percent for the leader and ten percent for the followers [2]. Moreover, trucks in a platoon occupy less space which means road utilization decreases and so does the likelihood of head-tail collisions. These benefits have made platooning the subject of heightened interest recently (see [3] for an overview of platooning projects). As a result, multiple field tests have been conducted or are planned around the world (see for example [4])

In the platooning literature, most research thus far has focused on technological and human factor issues. Recently, researchers have started looking into platoon planning from a transport and optimization perspective (see [2] for a recent review). Proper planning of platoons is required when the number of trucks equipped with technology is limited. For trucks to form a platoon, we need to synchronize their departure times and routes. For instance, it may be necessary for a truck to make a detour or depart a little later (or earlier) to join a platoon.

In the initial stages of platoon deployment on open roads, platoons are likely to be restricted to two trucks due to safety and legal reasons [1]. In this study, we focus on this setting.

2 The two-truck platooning problem

Consider a set of trucks in which each truck is associated with an origin, a destination, an earliest departure time, and a latest arrival time. The latest arrival time determines the time flexibility for a truck. The flexibility is the additional time a truck is able to spend as compared to the shortest time between its origin and destination. The source of this additional time might be a small detour to meet up with another truck or waiting for another truck.

The two-truck platooning problem involves forming platoons to minimize costs. For each possible two-truck platoon, this involves determining whether it is time feasible to platoon and, if so, finding the meeting and split points such that the arrival deadlines are met. A simple representation of a platoon is shown in Figure 1. Note that if the origins (destinations) of the

trucks are the same, there is no first (last) leg. We denote the percentage fuel saving in the platooning leg as compared to driving alone by $\rho \in (0, 1)$.

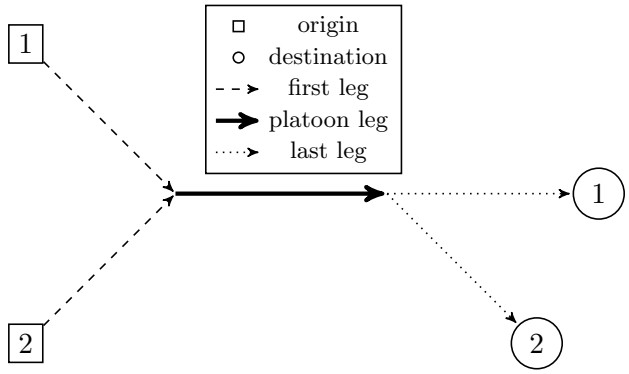


Figure 1: The three legs in a typical platooning trip

3 Solution approaches

Our approach to finding the best platoons consists of two steps. First, we create all the possible platoon matches from the set of trucks. In the second step, we represent the problem as a general matching problem by creating two identical sets of nodes with each node representing a truck. Two nodes (one from each set) are connected with an arc if the two represented trucks are distinct and may feasibly form a platoon. The weight of a connection denotes the cost (savings) of platooning. We maximize the number of matches (platoons) and then minimize total costs for this number of matches. Though it is true that the maximal platoon matching may not be cost optimal, we choose to maximize the number of platoons from the practical standpoint that it might be beneficial for the long term future of platooning to involve as many trucks (or companies) as possible (as we discuss in more detail in [2]). In this abstract, we describe different algorithms to create the set of possible platoons for a special case without time windows.

3.1 Creating optimal platoon matches

Due to the limited space available, we focus on the network case of platooning. In the full paper, we also consider platooning in Euclidean space.

To compute (possible) platoon benefits, we determine the optimal paths of trucks in the situation they were to meet. We can then compare the costs of platooning with the costs of the trucks travelling on their shortest paths.

3.1.1 Reducing the two-truck problem to a shortest path problem (SP)

In this exact approach, we formulate the two-truck platooning problem as a shortest path problem on an auxiliary graph. To create this auxiliary graph for a pair of trucks, we use the three parts of a route as shown in Figure 1. To the original graph, we add a source node and connect this source node to every other node by means of arcs. The weight of an arc from the source

node to a particular node is the sum of the shortest distances for each truck to get to that node from its respective origin. Then, the arc represents the sum of the first legs if both trucks meet at the node in consideration to form a platoon. We do the same for the last legs by connecting every node to a sink node with an arc whose weight is the sum of the shortest distances for each truck to get from that node to its respective destination. The arcs in the original graph may represent the platoon leg of the route and their weight is reduced by the fuel savings factor ρ. The shortest path between the source and the sink on the auxiliary graph represents the platoon route. The trucks meet at the node succeeding the source and split up at the node preceding the sink.

3.2 Heuristic approach to create platoons (H)

In this approach, we heuristically reduce the number of options to explore viable platoon pairs by fixing the route of one of the trucks to be its shortest path. From the original graph, we create a modified graph in which we reduce the cost of travelling on this path by the platoon savings factor. The arcs along this path then represent the platoon leg. We then find the shortest path of the other truck on this modified graph. If there is an overlap in the routes, the trucks may form a platoon along the overlap. We then reverse the roles of the truck and repeat this process again and choose the platoon (if any) with minimum costs. If there is no overlap in the routes, the trucks travel on their shortest paths.

3.2.1 Improvements to the heuristic approach

Improved platoon routes (H1). We fix the matches generated by the heuristic as platoon pairs. Only for each of these platoon pairs, we reduce the problem to a shortest path problem as described earlier to find the platoon route.

Hybrid approach (H2). We fix the matches generated by the heuristic as platoon pairs. For each platoon pair, we reduce the problem to a shortest path problem but with a smaller set of candidates for the meeting and split points. We consider the points on the routes generated in the heuristic (H) as candidates for the meeting and split points, instead of all the nodes in the graph.

4 Preliminary results

We use a real life instance with 100 truck trips to be routed on a network representing the street network of the Netherlands with 13112 nodes. We assume that the fuel savings factor ρ is 7%. Figure 2 shows the savings as a result of forming two-truck platoons for the special case without time windows. These results indicate that the heuristic along with its improvements have a good potential to perform well.

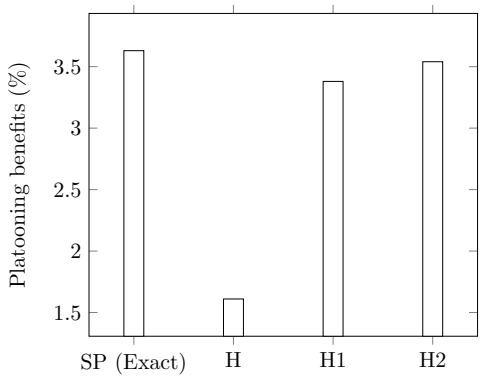


Figure 2: Preliminary results - platooning benefits

5 Work in progress

We are currently working on iterative approaches to incorporate time windows into our algorithms. In these approaches, we forbid certain meeting and split points if they are time infeasible and solve the problem further. We are presently testing these on smaller networks by comparing results to the optimal solution which we calculate by smartly enumerating over nodes in the network. In doing so, we are investigating the effect of introducing time windows on the performance of the heuristics by comparing the results to the case without time windows.

In addition to the approaches described above, we are building a heuristic that draws inspiration from our analytic approach for the two-truck platooning problem in Euclidean space. At the TRISTAN conference, we will present results from these and also from more experiments showing effects of various factors such as the number of trucks, percentage of trucks with platooning technology, varying network configurations etc. on platooning benefits.

References

[1] Robbert Janssen, Han Zwijnenberg, Iris Blankers, and Janied de Kruijff. Truck platooning: Driving the future of transportation, 2015.

[2] Anirudh Kishore Bhoopalam, Niels Agatz, and Rob Zuidwijk. Planning of truck platoons: A literature review and directions for future research. *Transportation Research Part B: Methodological*, 107:212–228, 2018.

[3] Carl Berenghem, Steven Schladover, and Erik Coelingh. Overview of platooning systems. In *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[4] Jacqueline Eckhardt, Loes Aarts, Arjan van Vliet, and Tom Alkim. European truck platooning challenge 2016, lessons learnt, 2016.

Network performance under different levels of ride-sharing: A simulation study

Negin Alisoltani [a,b]
14-20 Boulevard Newton, 77420 Champs-sur-Marne, France
Email: negin.alisoltani@ifsttar.fr

Ludovic Leclercq [a]

Mahdi Zargayouna [b]

[a] Univ. Lyon, IFSTTAR, ENTPE, LICIT

[b] University of Paris-Est, IFSTTAR, GRETTIA

1 Introduction

In recent years, intelligent transportation systems made it possible for operators to adapt in real-time the transportation supply to travel demand via new mobility services. Among these services, ride-sharing is becoming popular. Ride-sharing is a transportation mode in which passengers can share a car and travel costs. Dynamic ride-sharing refers to a system which supports an automatic ride-matching process between participants on very short notice or even en-route [1]. The dynamic ride-sharing problem involves two subproblems: 1) How to serve the upcoming trips (optimal fleet management) - 2) How to accurately predict the travel times to determine vehicles availability and pick up/drop off times.

The first subproblem is complex and has attracted a great number of research proposals ([2, 3, 4]). Following this track of research, we express the optimal fleet management problem as a constrained multi objective integer linear programming.

The second subproblem is less studied in the literature but is very important for real field operations. Network congestion can have significant impacts on the ride-sharing service. The optimization system of the ride-sharing service uses estimates for the predicted travel time coming from a "prediction model". When the rides are realized, a gap can exist between the estimation and the real traffic condition, that is represented by the "plant model". This gap may require dynamic adjustment of the initial assignment to fit with the observed conditions. When simulating a dynamic ride-sharing service, it is important to properly distinguish the prediction and the plant

model to propose a realistic solver.

In most of the researches, the plant model and the prediction model are the same [4, 5]. However, there are some researches that consider dynamic traffic conditions on ride-sharing. Goel et al. [4] consider an overhead randomly chosen of 10-20 percent to reflect different traffic conditions when computing the end time for a driver in their proposed approach. Nevertheless, They just use the prediction model and assume that the travel times used in the assignment process stay the same during the execution of the vehicle schedules. In some researches, only the plant model is considered. They use a simulator to assess the dynamic ride-sharing but it is not the optimal matching [5, 6, 7]. Other works use only static travel times in the optimization process [8].

In this paper, we define the plant model besides the prediction model to assess the impact of traffic conditions on the dynamic ride-sharing system performance for large-scale problems. The considered prediction model is based on the last observed travel times, while the considered plant model is a trip-based Macroscopic Fundamental Diagram (MFD) model which is able to reproduce the time evolution of mean traffic conditions for a full road network using the MFD as a global behavioral curve [9, 10]. In this paper, for a given urban network, we are going to compare the reference situation where all trips are done with personal cars with a situation where a fraction of the trips (market-share: 20%,60% and 100%) are served by a fleet of vehicles with different levels of maximum sharing (1,2,3) for all passengers.

2 Methodology

Our system has two main parts. The fleet management part works to assign the optimized match of riders to the vehicles. Then the simulation part executes the optimal car schedule while considering the complete dynamic traffic conditions.

We have defined an algorithm to find the optimized schedule for the shared cars. The algorithm solves a constrained optimization problem to minimize the total travel time and distance for vehicles and the total travel time and waiting time for passengers. The constraint functions in the problem are on capacity, time window, number of sharing (a number defined by the passengers to show their willingness to share their ride) and the quality of service. Let us to recall here that the optimization problem is solved with predicted travel time but then the simulation of the vehicle operation is matched with another model with a more refined description of the system dynamics.

To solve the optimization problem efficiently, we could refer to heuristic methods but we notice that a proper exploitation of the constraints can help to narrow the search of feasible solutions even if the size of the space is very large. This is why we design our own solution method based on the classical branch and bound algorithm but with specific properties to fit with fleet management problem.

A simulation platform is used as the plant model to simulate the function of both shared and personal cars. This simulator should be able to simulate the time evolution of traffic flows on the road network. In this research, we use the trip-based MFD to accommodate individual trips while keeping a very simple description of traffic dynamics. The general principle of this approach is to derive the inflow and outflow curves noting that the travel distance L by a driver entering at time $t - T(t)$ when $n(t)$ is the number of en-route vehicles at time t and the mean speed of travelers is $V(n(t))$ at every time t, should satisfy the following equation:

$$L = \int_{t-T(t)}^{t} V(n(s))ds \tag{1}$$

At each time step, the simulator computes the current speed of the cars considering the current traffic situation (the number of en-route vehicles). Then the vehicle can cover a distance based on the current speed, every time step (10 seconds). So the state of en-route cars is updated every 10 seconds in our simulation. To make travel time prediction for the optimization part, In our prediction model, we predict the traffic situation for the next assignment time horizon (every 10 minutes) and we assign the passengers to the cars based on this prediction.

3 Numerical experiments

In the proposed research, we use a realistic O-D trip matrix for the city of Lyon in France. The network is loaded with travelers of all ODs with given departure time in order to represent 4 hours of the network with more than 62000 requests based on the study of [11]. 23 different scenarios are defined with number of sharing 0,1,2 and 3 (Number of sharing 0 means that the car serves just one passenger without sharing like traditional taxi services, number of sharing 1 means that it is possible to share the passengers trip with 1 other passenger and so on), market shares 20%,60% and 100% (Only the trips that are fully inside the studied area are considered as candidates for the service. So the market share of 100% corresponds to 22% of all trips), two intervals for pick up and drop off time window (5 minutes and 10 minutes). Here, we put a part of results to show the system performance. Figure 2 shows the accumulation of cars in the network for different market shares when there is no sharing (like traditional taxi services). As the market share increases, the travel distance increases so the accumulation of cars moving in the system increases. In other words, the traffic conditions are deteriorated if we replace all the internal personal trips with service trips because idle vehicles are adding extra travel distances between two trips to be served.

Table 1 shows the cars travel time and number of cars for different numbers of sharing when the market share is 20 percent comparing with the case that all the trips are done with personal cars. Results show that with sharing, the number of cars and total travel time is less than the case without sharing or even the case with zero market rate.

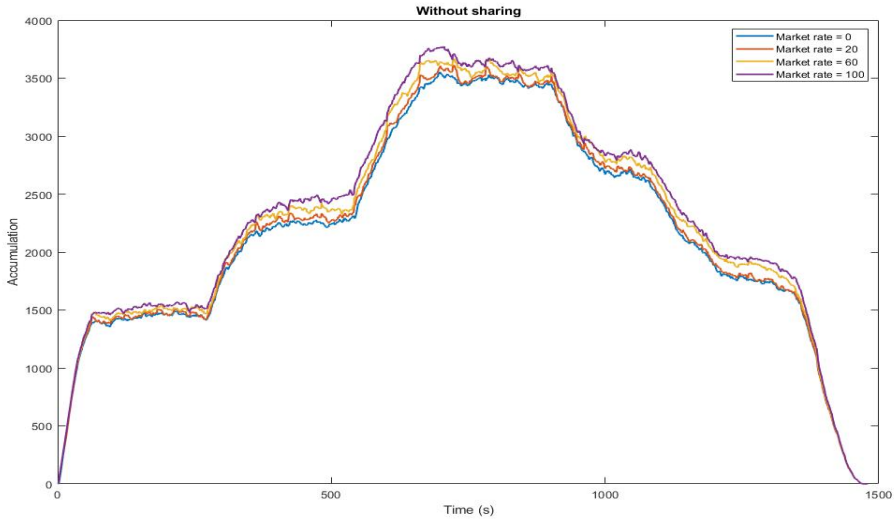


Figure 1: Cars accumulation for different market shares

Table 1: Simulations results for market-share = 20%

Sharing properties		Simulation results					
Market rate	Number of sharing	Total travel time for shared cars (s)	Total travel time for personal cars (s)	Total travel time for all cars	Number of shared cars	Number of personal cars	Total number of cars
0	0	0	38723249	38723249	0	62450	62450
20	0	1092250	37827210	38919460	2113	60214	62327
	1	993790	37749480	38743270	1184	60214	61398
	2	975050	37744950	38720000	992	60214	61206
	3	962170	37721130	38683300	929	60214	61143

When 20 percent of internal trips are served with service cars without any sharing, the number of needed service cars is 2113 and the total travel time for shared trips is 1092250 seconds. But then with sharing the ride between just two travelers, the number of needed cars decreases to 1184 and the total travel time is 993790. It means that with almost half number of cars, the travel time is 98460 seconds less than before. With applying more sharing, the number of needed cars and the total travel time decrease then for number of sharing 2 and 3, our proposed ride-sharing system works even better than the situation that all the trips are done with personal cars. It should be mentioned that the increase in passengers travel time and waiting time is negligible compared to the service improvements in our results.

In future researches, we will implement our system on larger networks with more number of trips. We will improve the optimization algorithm introducing spatial clustering on the network. Also we try to switch the plant model to a more refined one. Sensitivity analysis can be done on the system setting characteristics to select proper values based on the network and demand size.

References

[1] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[2] M. Ota, H. Vo, C. Silva, and J. Freire, "Stars: Simulating taxi ride sharing at scale," *IEEE Transactions on Big Data*, vol. 3, no. 3, pp. 349–361, 2017.

[3] X. Qian, W. Zhang, S. V. Ukkusuri, and C. Yang, "Optimal assignment and incentive design in the taxi group ride problem," *Transportation Research Part B: Methodological*, vol. 103, pp. 208–226, 2017.

[4] P. Goel, L. Kulik, and K. Ramamohanarao, "Optimal pick up point selection for effective ride sharing," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 154–168, 2017.

[5] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1782–1795, 2015.

[6] M. P. Linares, L. Montero, J. Barceló, and C. Carmona, "A simulation framework for real-time assessment of dynamic ride sharing demand responsive transportation models," in *Winter Simulation Conference (WSC), 2016*, pp. 2216–2227, IEEE, 2016.

[7] Y. Jia, W. Xu, and X. Liu, "An optimization framework for online ride-sharing markets," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pp. 826–835, IEEE, 2017.

[8] W. Herbawi and M. Weber, "The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1–8, IEEE, 2012.

[9] R. Lamotte and N. Geroliminis, "The morning commute in urban areas: Insights from theory and simulation," tech. rep., 2016.

[10] G. Mariotte, L. Leclercq, and J. A. Laval, "Macroscopic urban dynamics: Analytical and numerical comparisons of existing models," *Transportation Research Part B: Methodological*, vol. 101, pp. 245–267, 2017.

[11] J. Krug, A. Burianne, and L. Leclercq, "Reconstituting demand patterns of the city of lyon by using multiple gis data sources," tech. rep., University of Lyon, ENTPE, LICIT, 2017.

Investigating the robustness of route-based sensor location policies under variable network demand

Marco Rinaldi

Faculty of Science, Technology and Communication

University of Luxembourg

Email: marco.rinaldi@uni.lu

Francesco Viti

Faculty of Science, Technology and Communication

University of Luxembourg

1 Introduction

Traffic management applications rely on timely, precise and as complete as possible traffic flow information, in order to appropriately react to the road network's situation. Considering the sensing infrastructure required to collect said information, several approaches have been developed in order to determine both quantity and location of sensors required to reach a sufficient level of information, both in terms of quality and quantity [1].

Among others, the link flow inference problem leverages the algebraic relationships between different flows in a network, considering both node-link relations (conservation of flows at nodes) and link-route relations (conservation of vehicles at routes) [2]–[4]. These works are largely static in nature: optimal sensor locations are determined based on the network topology itself, without explicit consideration of changes in the network's behaviour due to the dynamic nature of transportation demand. Few works in literature have focussed on developing sensor location approaches that explicitly consider this variability, by means of stochastic optimisation [5]–[8], at an unavoidable loss in computational efficiency.

In this work we evaluate, through comparative analysis, how link flow inference-based sensor location approaches, albeit static in nature, behave when dealing with different demand levels. Specifically, our objective is assessing the amount and variability of estimation error induced by disregarding the stochasticity of demand when determining the optimal set of sensor locations.

By comparing two different static sensor location problem methodologies we showcase both how relevant the chosen static algorithm is, and quantify the effective information loss due to demand variability.

2 Methodology

Variations in the volume of traffic demand induce considerable changes in the user's preferred route set: when overall demand is very low, users will choose the topologically shortest path to their destination, disregarding any other, longer alternative. As demand rises, the formation of congestion pushes users towards other alternatives, in order to maintain their own perceived cost as low as

possible [9]. In link flow inference problems, route information is assumed fixed and static, and the resulting sensor locations are largely related to which routes are included in the chosen set [10].

In this work we compare the impact of two different route set enumeration policies, namely the simpler K-Shortest Path [11] and our recently developed hypergraph-based approach [12], and evaluate how the sensor locations determined through this static selection of routes compares to those dynamically arising from deterministic assignment. Full observability solutions are obtained for both route enumeration approaches using Castillo's Pivoting technique [13]. The overall comparative approach is summarised in Figure 1.

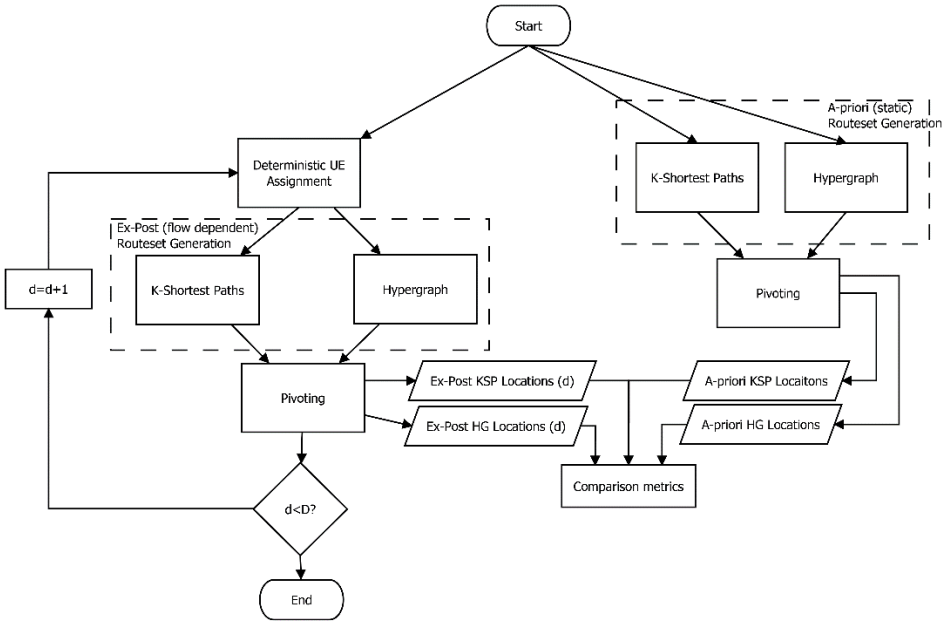


Figure 1: Flowchart of this work's comparative approach

Both a-priori sensor location sets $\Omega_{KSP}, \Omega_{HG}$ are determined based solely on topological network costs, whereas the ex-post counterparts $\Omega_{KSP}(d), \Omega_{HG}(d)$ are determined considering the link travel costs arising from Dial's B deterministic traffic assignment procedure [cit dial]. Repeated assignment is carried out considering a base Origin-Destination demand matrix $X_b \in \Re^{n \times n}$, which is gradually multiplied by an amplitude modifier $\alpha = [\alpha_1, ..., \alpha_D]$. This implies that, rather than considering variations in the spatial distribution of demand, we are focussing, in these preliminary results, in demand amplitude variations, and how these affect the overall user's route choice.

Demand-dependent cross-comparison is carried out considering three indicators:

- The total amount of sensors necessary to fully observe the given network;
- The percental overlap between the sensors resulting from the a-priori approaches and those of the ex-poste approach
- The partial observability level resulting by locating only sensors according to the a-priori approaches, as measured by the NSP metric (eq. 1)

$$NSP(\Omega_*(\alpha)) = \frac{\| \Omega^T{}_*(\alpha) B_*' \|_F}{\| \Omega^T{}_*(\alpha) \|_F} \quad (1)$$

where $\Omega_*(\alpha)$ is the full observability matrix pertaining to the ex-post solution, while B_*' is the partial observability solution obtained by considering the set of sensors in the intersection $\Omega_*(\alpha) \cap \Omega_*$, that is, those sensors pertaining both to the a-priori and the ex-post solution.

3 Experimental results

We apply our cross-comparison on a simplified version of the road network pertaining to the Dutch city of Rotterdam, including its Ring Road and the main surrounding motorway accesses, as shown in Figure 2.

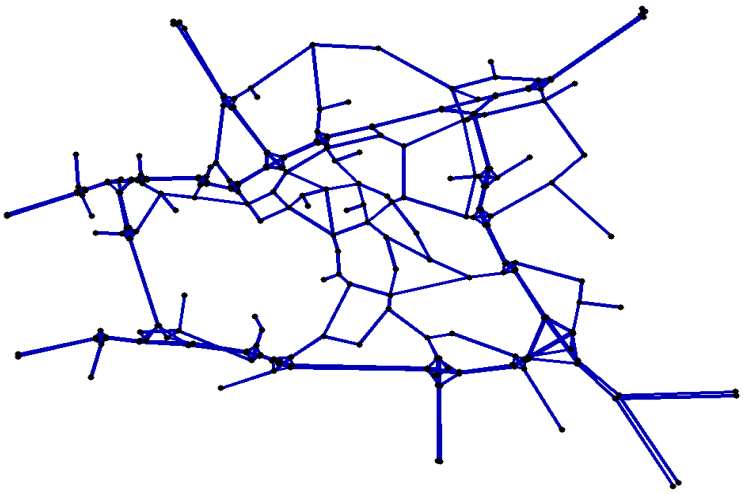


Figure 2: The Rotterdam road network.

Demand representing morning peak conditions is used as the base scenario, we consider multiplicative factors $\alpha = [0.1, ..., 3]$ with steps of 0.1. The three comparative metrics discussed above are showcased in Figure 3.

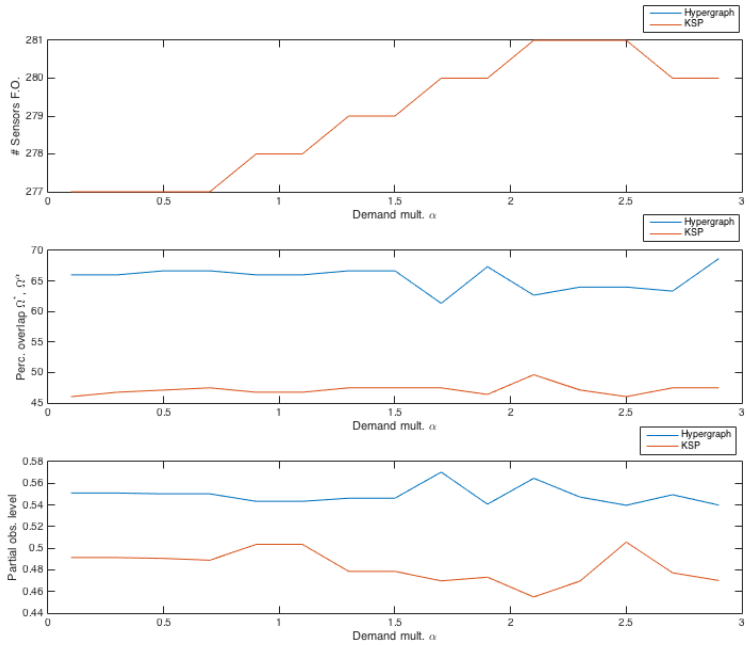


Figure 3: Test results for the three chosen comparison metrics

Interestingly, while the total amount of sensors required to fully observe the network increase with demand (which is rather expectable, as higher demand levels directly imply a more widespread usage of the network), this quantity is independent of the chosen route set generation approach. Conversely, considerable differences arise both in terms of percental overlap between a-priori and ex-post sensor locations, and resulting partial observability level. Indeed, the hypergraph generated approach, due to its inherent higher level of prior information, is an overall better candidate than the standard K-Shortest Path approach, even for varying levels of demand. From this preliminary analysis we can anyhow conclude that, using the better approach, information loss due to route choice mismatch reaches an average level of 40%, attesting to the fact that while static solutions can be lossy, a considerable amount of information on link flows can still be extracted successfully.

Further comparison results, considering variations not only in the amplitude of demand, but also on its geographical distribution, will be presented at the symposium.

References

[1] E. Castillo *et al.*, "A State-of-the-Art Review of the Sensor Location, Flow Observability, Estimation, and Prediction Problems in Traffic Networks," *J. Sens.*, vol. 2015, p. e903563, Oct. 2015.

[2] M. Ng, "Synergistic sensor location for link flow inference without path enumeration: A node-based approach," *Transp. Res. Part B Methodol.*, vol. 46, no. 6, pp. 781–788, Jul. 2012.

[3] E. Castillo, A. Cobo, F. Jubete, R. Pruneda, and C. Castillo, "An Orthogonally Based Pivoting Transformation of Matrices and Some Applications," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 3, pp. 666–681, Jan. 2001.

[4] S. He, "A graphical approach to identify sensor locations for link flow inference," *Transp. Res. Part B Methodol.*, vol. 51, pp. 65–76, May 2013.

[5] C. Fu, N. Zhu, and S. Ma, "A stochastic program approach for path reconstruction oriented sensor location model," *Transp. Res. Part B Methodol.*, vol. 102, pp. 210–237, Aug. 2017.

[6] X. Zhou and G. F. List, "An Information-Theoretic Sensor Location Model for Traffic Origin-Destination Demand Estimation Applications," *Transp. Sci.*, vol. 44, no. 2, pp. 254–273, Apr. 2010.

[7] N. Wang, M. Gentili, and P. Mirchandani, "Model to Locate Sensors for Estimation of Static Origin-Destination Volumes Given Prior Flow Information," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2283, pp. 67–73, Nov. 2012.

[8] X. Xu, H. K. Lo, A. Chen, and E. Castillo, "Robust network sensor location for complete link flow observability under uncertainty," *Transp. Res. Part B Methodol.*, vol. 88, pp. 1–20, Jun. 2016.

[9] J. G. Wardrop, "ROAD PAPER. SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH.," *ICE Proc. Eng. Div.*, vol. 1, no. 3, pp. 325–362, Jan. 1952.

[10] M. Rinaldi, F. Corman, and F. Viti, "Assessing the Effect of Route Information on Network Observability Applied to Sensor Location Problems," *Transp. Res. Procedia*, vol. 10, pp. 3–12, 2015.

[11] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Manag. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.

[12] M. Rinaldi and F. Viti, "Exact and approximate route set generation for resilient partial observability in sensor location problems," *Transp. Res. Part B Methodol.*, vol. 105, no. Supplement C, pp. 86–119, Nov. 2017.

[13] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Traffic estimation and optimal counting location without path enumeration using Bayesian networks," *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 23, no. 3, pp. 189–207, 2008.

Truck Platooning Network Design

Szymon Albiński, Stefan Minner
School of Management,
Technical University of Munich
Email: szymon.albinski@tum.de

Teodor Gabriel Crainic
CIRRELT & School of Management,
Université du Québec à Montréal,
Montréal, Canada

1 Introduction

Advances in autonomous driving technology have fostered the idea of truck platooning. Thereby, several trucks drive in close succession, connected by a data link, thus exploiting the predecessor's slipstream. This allows for fuel savings by up to six percent. An additional savings potential implies a proposal currently debated by the European Union. The bill suggests that the time spent in the platoon partially counts as rest time for the drivers in the following trucks. Hence, less stopovers are needed, which would help to cut cost as well as to reduce the issues pertaining overcrowded parking spots.

For carriers, truck platooning is most beneficial if they are willing to travel together with others. Thus, we assume that all carriers in the system share their trip information (origin and destination, earliest start and latest finishing time) with a central platform. Out of the registered trips, the platform creates platoons and returns information to the carriers if a trip is accepted and what the savings will be. A portion of the savings is kept by the platform as reward for the planning. Furthermore, the platform provides the carriers with individual routes and schedules for their tour.

Since we look at long distance networks like the European or the North American highway system, a planning period may cover several days or even up to one week. As a result of this long planning period, we distinguish between two types of trips: On the one side *regular trips*, which are planned some days or weeks in advance (e.g. factory traffic, inter-hub traffic of parcel services), on the other side *ad-hoc* trips that are dispatched on a short notice (e.g. transports of seasonal products). To provide planning security, the platform needs to give a reply to the regular trips a sufficient time in advance. However, ad-hoc trips may allow for additional or bigger platoons and thus for more savings. Hence, the platform operator could consider reserving capacities for ad-hoc trips when planning the regular trips. The platform is compensated for this risk by returning less savings for ad-hoc trips and by keeping a portion of additional savings that follow from possible

new platoons that are formed thanks to ad-hoc trips. This motivates us to formulate the following research question: *How to set-up a profitable service network for truck platoons that combines the off-line platooning of regular trips with the real-time platooning of ad-hoc trips?*

To plan the platoons formed by regular trips, we use a service network design model where we reserve spots for ad-hoc trips by assigning a value for reservation. This value is regularly updated based on historical information about ad-hoc trips. As soon as the ad-hoc trips are revealed, the actual platoons need to be planned on an operational level. Thereby, we have to guarantee the savings for all trips that were already accepted in the first stage.

The main contributions of our work are the following: (1) We propose a model that allows the formation of truck platoons that consists of regular and ad-hoc trips, (2) we develop a solution method that allows the application of the model to real-life settings, (3) based on our numerical study, we provide insights on the value of information, the influence of ad-hoc trips on the profitability of the network and different influencing parameters.

2 Problem description and assumptions

We assume that there exists a central platform, which coordinates the formation of platoons. Shippers can register their trips for a certain time period $T = [T^s; T^e]$ at this platform. All trips that are announced up to a registration time $T^r < T^s$ are called *regular*. Trips registered in the interval $[T^r; T^s)$ are called *ad-hoc*. Since every trip is associated with a truck, we refer in the following to trucks and denote the set of all trucks by $\mathcal{K}$. To reserve capacities for ad-hoc trips when building the platoons at T^r, the platform operator can add a subset $\mathcal{K}^{vir}$ of "virtual trucks" to $\mathcal{K}$. We introduce a node set $\mathcal{V}$ that includes (i) the origins and destinations of the trucks, (ii) the parking lots along the highways and (iii) *forming nodes*, which are virtual points that represent the highway entrances and exits. We name the last two type of nodes *waypoints* and collect them in the subset $\mathcal{V}_W \subsetneq \mathcal{V}$. We say that platoons can only be formed or disbanded at those waypoint nodes. The objective is to reduce the total travelling cost for all registered trucks. We assume that there exists a mechanism that fairly distributes the cost between the trucks within the same platoon.

To synchronize the schedules of the trucks, we divide the planning period T into t finer intervals of equal length (e.g. 15 minutes). To track the movements of the trucks, we use a space-time network where we expand each node $v \in \mathcal{V}$ t times. The node set V_K contains the time-expanded *truck nodes*, that is: $|V_K| = t \cdot |\mathcal{V}|$. We presume that the platoon size will be limited to s_{max} trucks, which results in $(s_{max} - 1)$ different platoon sizes. Since we need to distinguish platoons also by their size, we have to expand $\mathcal{V}_W$ in the platoon layer not only in time but also in in the platoon size. We call V_P, the set of these expanded nodes, *platoon nodes* and it holds that $|V_P| = (s_{max} - 1) \cdot t \cdot |\mathcal{V}_W|$. We interpret a platoon as

a direct service that "transports" a certain number of trucks from one point to another at a given cost at a given time. Thus, in this space-time-size network, the origin node $o_p \in V_P$ and the destination node $d_p \in V_P$ define at which waypoint and at which time platoon $p \in P$ starts and ends and how many trucks in this platoon are included. Note that it is possible that there are several platoons of type p. However, we assume that highway capacities limit this number to $\nu_p \in \mathbb{N}$.

The nodes in V_K form the *truck layer*, the nodes in V_P the *platoon layer*. Together with the arc set A, $V = V_K \cup V_P$ describes a graph $G(V, A)$. We partition A into *truck arcs*, *interlayer arcs* and *platoon arcs*. The truck arcs interconnect nodes on the truck layer, the platoon arcs nodes within the platoon layer. Trucks can wait at physical nodes, that is at their origins and destinations as well as at parking lots. To model this option, we use *waiting arcs* that connect a node in the truck layer to the identical physical location one time step later. Since every truck has a time window $[e_k; l_k]$, we create only those arcs that leave the truck's origin earliest at e_k and enter the destination not later than at l_k. Movements within the truck layer and the platoon layer are represented by *travel arcs*. The construction of those arcs is based on $\tau_{ij}(T^r)$, the traveltime forecasts for nodes $i, j \in \mathcal{V}$ at time T^r. The truck layer and the platoon layer are linked via interlayer arcs. These arcs connect only truck-waypoints and platoon-waypoints that are identical in location and time period. An interlayer arc from a truck-waypoint-node to a platoon-waypoint-node with size s represents the formation of a platoon of size s at the specific location and time period. Similarly, an interlayer arc leading from a platoon-waypoint-node with size s to the truck-waypoint-node means that a platoon of size s is disbanded at this location at the specified time interval. Consequently, every truck-waypoint is connected bidirectionally to the $(s_{max} - 1)$ platoon-waypoints that are identical in location and time.

A cost is assigned to every arc. The truck travel arcs are associated with the full cost of driving, c_{ijk}. Among other things, these cost include fuel expenses, wages, tolls and depreciation and they depend on the truck k. c_p denotes the cost associated with the corresponding platoon arc. We assume that this cost is lower than it would be if all participants were to travel the same route at the same time individually. Furthermore, the cost share per truck is indirectly proportional to the platoon size. Interlayer arcs are associated with cost $c_{i,o_p,k}$ for forming a platoon or leaving a platoon, $c_{d_p,j,k}$. These cost also allow to even out cost differences in the platoons due to different truck types. Truck waiting arcs have zero cost. For that reason, we use the convention that truck k's origin node $o_k \in V_K$ is the node that represents the truck's physical origin at time interval $t = T^s$. By the same convention, we assume that the destination node $d_k \in V_K$ is associated with the physical destination at time T^e. For all other arcs (which do not fulfil the criteria stated above), we set the travel cost to infinity.

3 Model formulation

We propose a mixed-integer linear program for planning the regular trips. It is defined on the space-time network G that we discussed above. We introduce two decision variables. The binary decision variable x_{ijk} is set to 1 if truck k uses arc (i, j), $i, j \in V$. The integer decision variable y_p counts how many platoons p travel on arc (o_p, d_p), $o_p, d_p \in V_P$. s_p denotes the size of platoon p. The *Platoon Network Design Problem* (*PNDP*) reads as follows:

$$\min \sum_{p \in P} c_p y_p + \sum_{k \in \mathcal{K}} \sum_{i,j \in V} c_{ijk} x_{ijk} \tag{1}$$

$$s.t. \quad \sum_{j \in V_K} x_{o_k,j,k} = 1 \qquad \forall k \in \mathcal{K} \tag{2}$$

$$\sum_{i \in V_K} x_{i,d_k,k} = 1 \qquad \forall k \in \mathcal{K} \tag{3}$$

$$\sum_{j \in V_K} \left(x_{ijk} - x_{jik} \right) = 0 \qquad \forall i \in V_K \backslash \{o_k; d_k\}, k \in \mathcal{K} \tag{4}$$

$$\sum_{k \in K} \sum_{i \in V_K} x_{i,o_p,k} = s_p y_p \qquad \forall p \in P \tag{5}$$

$$x_{i,o_p,k} - x_{d_p,j,k} = 0 \qquad \forall i, j \in V_K, k \in \mathcal{K}, p \in P \tag{6}$$

$$y_p \leq \nu_p \qquad \forall p \in P \tag{7}$$

$$x_{ijk} \in \{0; 1\}, y_p \in \mathbb{N}_0 \qquad \forall i, j \in V, p \in P, k \in \mathcal{K} \tag{8}$$

Equalities (2) and (3) ensure that each truck leaves his origin and enters his destination. For all other nodes in the truck layer, (4) conserves the flow. Equation (5) states that a platoon can only be formed if the correct number of trucks meets at the waypoint and according to (6), a truck that joined a platoon has to leave it at the platoon's destination. The number of platoons p is limited by (7). In the objective function (1), the total cost of travelling in the network is minimized.

4 Conclusion

We present an approach that brings together the tactical and operational planning and scheduling of truck platoons. For the tactical planning, we use a space-time network and include the possibility of reserving capacities for possible ad-hoc trips that may be announced on short notice. We want to develop a planning tool for the practical use and therefore we aim at achieving good solutions in reasonable time. For that purpose we focus on developing a matheuristic. This heuristic as well as further details on the inclusion of the ad-hoc trips will be presented together with the numerical results during the conference.

A Dynamic Ride-Sourcing System for City-Scale Networks

Amir Hosein Valadkhani

The University of Sydney, School of Civil Engineering, Sydney, Australia

Mohsen Ramezani

The University of Sydney, School of Civil Engineering, Sydney, Australia

Email: mohsen.ramezani@sydney.edu.au

1 Introduction and Motivation

Ride-sourcing systems (e.g., traditional taxis) are a popular mode of transportation because of their convenience and recent affordability caused by emerging on-demand services that use mobile apps as a platform to match passengers with vacant taxis. These new platforms enable the service provider to implement dynamic matching and transfer of ride-sourcing vehicles to increase the system's profit while ensuring a satisfactory level-of-service such as keeping the expected passenger waiting time below a threshold. The properties of the ride-sourcing system are investigated according to the minimum number of the ride-sourcing fleet [1, 2], improving the network delay [3], or reducing the travel time for ride-sourcing customers [4]. In this work, we introduce a ride-sourcing system to reduce the waiting time for ride-sourcing customers and search time of the ride-sourcing vehicles, and to improve the trip travel time.

To this end, we introduce a ride-sourcing method for matching ride-sourcing vehicles and passenger's travel requests while transferring unmatched ride-sourcing vehicles to the regions with excess of unmatched passengers to balance the supply and the demand of the ride-sourcing market. The method consists of two layers: (i) *matching layer* that is responsible for matching the passengers to the unmatched ride-sourcing vehicles and (ii) *transfer layer* that transfers the unmatched ride-sourcing vehicles to the regions with excess of unmatched passengers. The effectiveness of the proposed method is tested in a microsimulation case study.

2 Methodology

2.1 Framework Description

The proposed dynamic ride-sourcing system consists of the top layer (i.e. matching layer) and the bottom layer (i.e. transfer layer), as illustrated in figure 1. The matching layer includes two methods: (i) *matching* method and (ii) an *adaptive rejection* method. The matching method finds the optimum, in terms of total matching distance, allocation of the unmatched ride-sourcing vehicles and unmatched passengers. The adaptive rejection method declines the matchings where the pickup distances are longer than an adaptive threshold. The transfer layer consists of an optimization and transfer allocation methods. The first method derives the optimum number of the transferred ride-sourcing vehicles to be requested to move (without any passenger) from their current region to another region to address the shortage of ride-sourcing vehicles and excess of waiting passengers. The latter method determines the set of individual unmatched ride-sourcing vehicles that are selected to be transferred.

The urban network is assumed to be partitioned into different regions based on homogeneity of traffic states (e.g. passenger arrival). The matching layer acts at the network level and very frequently (e.g. in order of seconds) and the transfer layer transfers the unmatched ride-sourcing vehicles between regions less frequently (e.g. in order of minutes). The matching layer is triggered every k^{m} time step. This layer obtains the position of individual unmatched ride-sourcing vehicles, $c^{\mathrm{um}}(k)$, position of unmatched passengers, $p^{\mathrm{um}}(k)$, and the position of transferred ride-sourcing vehicles, $c^{\mathrm{t}}(k)$, at time step k. It allocates the unmatched passengers to the union of transferred and unmatched ride-sourcing vehicles.

The transfer layer is triggered every k^{t} time step ($k^{\mathrm{t}} > k^{\mathrm{m}}$). The optimization method collects the total number of the transferred and unmatched ride-sourcing vehicles, and unmatched passengers in region i from the plant and matching layer to return the total number of the desired transferred regional ride-sourcing vehicles, $\|\tilde{c}_i^{\mathrm{t}}(k)\|$. The transfer allocation determines which unmatched ride-sourcing vehicles in region i must be transferred to region j, $i \neq j$.

2.2 Matching Layer

The optimum matching between unmatched ride-sourcing vehicles and waiting passengers is determined by solving the minimum weighted matching problem for a bipartite graph. We construct the problem as a bipartite graph by considering, (i) V_1 as the set of transferred and unmatched ride-sourcing vehicles, (ii) V_2 as the set of the waiting passengers, and (iii) E as the edges connecting each element of the V_1 to V_2 (V_1 and V_2 are disjoint and independent sets). The weights of the E are the distance between the elements of the V_1 and V_2. We obtain the minimum weighted

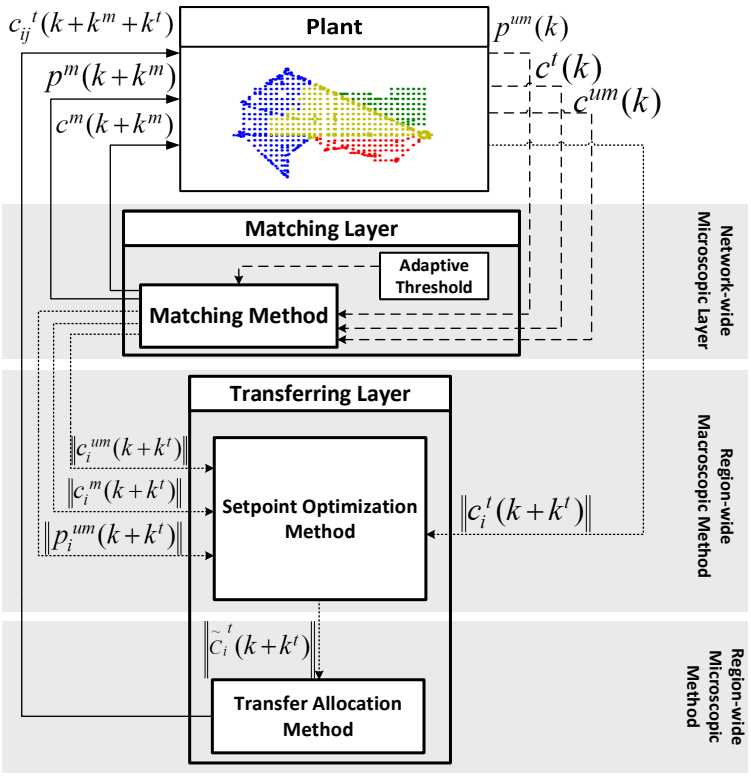


Figure 1: Proposed hierarchical ride-sourcing system

matching for this bipartite graph via integer linear programming method:

$$\min \quad \sum_{e \in E} x_e w(e), \tag{1}$$

$$\text{s.t.} \quad \sum_{e \sim v} x_e \leq 1 \quad \forall v \in \{V_1 \cup V_2\} \quad \& \quad x_e \in \{0,1\} \quad \forall e \in E, \tag{2}$$

where, $w(e)$ is the weight of each edge $e \in E$ and $e \sim v$ denotes e is an incident on v.

The adaptive threshold for rejecting the matching is formulated as:

$$\delta(k) = A \frac{\nu(k)}{\bar{\rho}}, \tag{3}$$

where, A is a constant scaling factor. $\nu(k)$ and $\bar{\rho}$ denote the average network speed at time step k and average probability for appearing new passengers in the urban network. The matches between ride-sourcing vehicles and waiting passengers that requires the pick up distance to be longer than $\delta(k)$ are discarded and the unmatched vehicles and passengers remain in the system to be matched at the next time interval, i.e. $k + k^{\mathrm{m}}$.

2.3 Transfer Layer

The transfer layer balances the demand and supply of the ride-sourcing system within each region. The boarding function that estimates the number of the boardings (actual pickups of passengers

by ride-sourcing vehicles) in each region is assumed as a Cobb-Douglas form with time-invariant elasticities. Hence, the optimum number of the transferred ride-sourcing vehicles to maximize the number of the boarding in all regions is obtained via:

$$\max \left(\sum_{i=1}^{N} \alpha_i \|p_i^{\mathrm{um}}(k)\|^{\beta_i} \left(\|c_i^{\mathrm{m}}(k)\| + \|c_i^{\mathrm{t}}(k)\| + \|\tilde{c}_i^{\mathrm{um}}(k)\| \right)^{\gamma_i} \right), \tag{4}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \|\tilde{c}_i^{\mathrm{um}}(k)\| = \sum_{i=1}^{N} \|c_i^{\mathrm{um}}(k)\| \quad \& \quad \|\tilde{c}_i^{\mathrm{um}}(k)\| \geq 0, \tag{5}$$

$$\rightarrow \quad \|\tilde{c}_i^{\mathrm{t}}(k)\| = \|\tilde{c}_i^{\mathrm{um}}(k)\| - \|c_i^{\mathrm{um}}(k)\|, \tag{6}$$

where, β_i and γ_i are elasticities with respect to unmatched passengers and vacant ride-sourcing vehicles for region i. α_i denotes the total productivity factor of region i and N is the number of the regions.

The transfer allocation method solves the minimum weight problem for the bipartite graph with integer linear programming. The two independent sets of the graphs are: (i) all individual unmatched ride-sourcing vehicles in regions with extra unmatched passengers and (ii) all the candidate location in regions with extra unmatched passengers. The weights of the edges are the distance between unmatched ride-sourcing vehicles and the locations.

3 Preliminary Results

We utilize the calibrated microsimulation model of the city center of Barcelona to evaluate the proposed ride-sourcing method. In Table 1, simple distance-based matching scenario matches the ride-sourcing vehicles with passenger's travel requests based on minimizing the total distance without considering the adaptive threshold and transfer layer.

Table 1: Assessment of the proposed ride-sourcing system

	Matching Travel Distance [km]		**Search Travel Time [min]**		**Passenger Trip Time [min]**		**Passenger Waiting Time [min]**	
	Total	**Mean**	**Total**	**Mean**	**Total**	**Mean**	**Total**	**Mean**
Distance-Based Matching	1916	0.93	13792	6.67	17722	8.58	13107	6.34
Proposed Ride-Sourcing	1214	0.59	12067	5.88	17211	8.38	9802	4.78

Matching travel distance is the distance that matched ride-sourcing vehicles traverse to pick up a matched passenger. Search travel time is the time that unmatched ride-sourcing vehicles spend to pickup a passenger. Passenger trip time is the time that passengers are in a ride-sourcing vehicle. Passenger waiting time refers to the time between the passenger's travel requests and

the time she/he board a ride-sourcing vehicle. Total number of the boardings by applying the simple distance-based matching and the proposed method are 2066 and 2057. Table 1 shows the effectiveness of the proposed method to reduce the search time of vacant ride-sourcing vehicles and waiting time of passengers.

References

[1] X Zhan, X Qian, SV Ukkusuri, "A Graph-based Approach to Measuring the Efficiency of an Urban Taxi Service System", *IEEE Transactions on Intelligent Transportation Systems* 17, 2479-2489 (2016).

[2] MM Vazifeh, P Santi, G Resta, SH Strogatz and C Ratti, "Addressing the Minimum Fleet Problem in On-demand Urban Mobility", *Nature* 557, 534 (2018).

[3] M Ramezani and M Nourinejad, "Dynamic Modeling and Control of Taxi Services in Large-scale Urban Networks: a Macroscopic Approach", *Transportation Research Part C: Emerging Technologies* 94, 203-219 (2017).

[4] Z Liu, T Miwa, W Zeng, M Bell, "Shared Autonomous Taxi System and Utilization of Collected Travel-Time Information", *Journal of Advanced Transportation* 2018, (2018).

Connected Vehicle Sensor Location Model for Traffic Congestion Mitigation

Hyoshin Park, Corresponding Author
Visiting Professor, NASA Jet Propulsion Laboratory, California Institute of Technology
Assistant Professor, Department of Computational Science & Engineering
North Carolina A&T State University, Email: hpark1@ncat.edu

Ali Haghani
Professor, Department of Civil and Environmental Engineering
University of Maryland

1 Introduction

Traditional studies related to sensor positioning have focused upon locating sensors to enhance the quality of traffic origin-destination (OD) demand [1] or travel time [2] estimations. This paper integrates automated traffic signal operation and sensor location problem in a connected vehicle environment with advanced data analytics. As a variant of mobile facility location problem, this study optimally allocates road-side sensors connected to traffic signal controllers to extend green light to prevent queue spillback, considering the future predicted delay of each intersection over the course of the day. Although previously developed author's two-stage stochastic programming model [3] provides scenario-based solutions with better performance than deterministic model, the high relocation cost has made researchers overlook the benefit of dynamic sensor relocation. With scheduling of autonomous robots, the synchronously commanding robots, drones, autonomous vehicles will present a reliable performance.

To improve approximation methods proposed in previous research [4], we employ heuristic algorithms to solve the proposed combinatorial optimization problem. Lagrangian relaxation decomposes the problem into two subproblems. Within feasible solutions provided by the first subproblem (relocation problem), the second subproblem (location problem) is solved until best bound is found. Cutting plane method adds a valid cut to the subgradient algorithm with better bound. To remedy the convergence problem in the subgradient algorithm, the location problem is solved with a variable neighborhood search method.

2 Anticipatory Dynamic Sensor Location Problem (SLP)

While previous signal optimization depends on the available resources [5], this study uses limited resources by relocating sensors. Until the increase in the penetration rate reaches a certain point, the transportation authority may be reluctant to relocate sensors because the rewards are low. By restricting the relocation frequency to once per sensor, we can have a partially anticipatory assumption. In this restricted problem setting [6], once one sensor is relocated, no more relocation can occur to that sensor. The formulation is simplified by assuming that there is no linkage between demand realizations and location decisions between some time periods. The independence assumption enables us to rewrite the multi-stage stochastic programming as a large two-stage stochastic programming. This assumption greatly reduces the complexity of the problem to solve much larger and more realistic instances.

We introduce a new auxiliary variable $z^t_{(i)}$ be equal to 1 if node i has a new sensor installed, -1 if a sensor at node i is relocated to another location, and 0 if there is no relocation. The vector difference of location is expressed as the sum of relocation variables $y^t_{(j)(l)}(j, l \in \mathcal{N})$ that is equal to 1 if there is a relocation from location j at time t to location l at time $t+1$.

$$\sum_j y^t_{(j)(l)} - \sum_l y^t_{(j)(l)} = z^t_{(i)} \qquad \forall i \in \mathcal{N}, \forall t \in \mathcal{T} \tag{1}$$

We enforce that there is no more sensor removal can occur when $z^t_{(i)} = -1$, and sensor cannot be installed at a location with an existing sensor when $z^t_{(i)} = 1$. Let $\boldsymbol{z}$ be a decision vector, then a sequence of $z^t_{(i)}$ for all time periods $t \in \mathcal{T}$ can be defined as $\left[z^1_{(i)}, \ldots, z^{\mathcal{T}}_{(i)}\right]$. The frequency of $\boldsymbol{z} = \left\{-1, 1\right\}$ is restricted to less than once for given operation period $\mathcal{T}$ as follows:

$$|\boldsymbol{z} = -1, 1| \leq 1 \qquad \forall i \in \mathcal{N} \tag{2}$$

We replace these relocation associated constraints and presents the multi-period dynamic SLP with restricted relocation.

$$\textbf{SLP} \quad \max_{\mu} \mathbb{E}_{\xi^1, \xi^2, \ldots, \xi^{\mathcal{T}}} \left[\sum_{t=1}^{\mathcal{T}} \psi^t(\boldsymbol{x}^t, \xi^t) \right] \tag{3}$$

$$s.t. \quad \begin{cases} \boldsymbol{x}^1 = \mu(\xi^1); \\ \boldsymbol{x}^t = \mu(\boldsymbol{x}^{t-1}, \xi^t) \quad \forall 2 \leq t, \quad t \in \mathcal{T}; \end{cases} \tag{4}$$

$$\sum_{i \in N} \boldsymbol{x}^1 \leq c \quad \forall t \in \mathcal{T}; \tag{5}$$

$$\left[\boldsymbol{y}\boldsymbol{\pi}\right]^t_{(j)(l)} \leq b^t_{(2)} \quad \forall t \in \mathcal{T}, \quad j = l; \tag{6}$$

$$b^t_{(2)} = b^{t-1}_{(2)} - \left[\boldsymbol{y}\boldsymbol{\pi}\right]^t_{(j)(l)} + b^t_{(1)} \quad \forall t \in \mathcal{T}; \tag{7}$$

$$x^t_i \leq \sum_{j=1}^{N} y^{t-1}_{(j)(l)} \quad \forall i; \tag{8}$$

$$|\boldsymbol{z} = -1, 1| \leq 1 \qquad \forall i \in \mathcal{N} \tag{9}$$

$$\boldsymbol{x}^t \in \{0, 1\} \quad \boldsymbol{y}^t \in \{0, 1\} \quad \forall t; \tag{10}$$

3 Solution Method

To solve large instances of dynamic sensor location problem, we enhance the solution efficiency through decomposition. We introduce a tight Lagrangian bound and an efficient dual heuristic that embedded a search heuristic. We will show that even with a reduced number of sensors, fair delay savings are guaranteed under feasible relocations. After reaching the maximum efficiency of the relocation, the level of diminishing marginal delay savings will become identical to a model without relocation. Since we cannot solve our SLP with submodular function, we introduce Lagrangian relaxation.

Lagrangian relaxation: First, we solve relocation problem to provide initial solutions with feasible links between optimal locations in each time period. Second, by fixing feasible links on the tree, the problem is simplified to find a reduced set of locations with some fixed locations defined by future relocations. Applying a relaxation guided variable neighborhood search to the reduced problem instances yields significantly better solutions in shorter time than when applying these metaheuristics to the original instances. We introduce lagrangean relaxation to separate the problem into two. Then, cutting plane algorithm is introduced to solve Lagrangian dual problem, and we move into the search heuristic.

The decomposition of $L(\lambda) = L_1(\lambda) + L_2(\lambda)$ will offer significant computational advantages over the original formulation. $L_1(\lambda)$ is calculated by solving solely relocation problem. $L_2(\lambda)$ is calculated by scenarios only the first stage influencing the rest of the stages. $\boldsymbol{x}^{\Lambda}$ will be a reduced set of location decision vector that has future relocations defined. The term of $L_2(\lambda)$ can be replaced by $\sum_{t=1}^{\mathcal{T}} \mathbb{E}_{\xi^1,\xi^2,...,\xi^{\mathcal{T}}} \left[\psi^t(\boldsymbol{x}^{\Lambda}, \xi^t)\right]$ with fixed relocations. Since we find an upper bound for each value of λ, as long it is nonnegative, we want to find the value of λ that leads to the tightest upper bound. We define this as Lagrangean dual problem as $min_{\lambda \geq 0} L(\lambda)$. By calculating the optimal solutions of two subproblems $\bar{\boldsymbol{x}}_i^t$ and $\bar{\boldsymbol{y}}_j^t$, we can solve $L_1(\lambda)$ and $L_2(\lambda)$. A subgradient of $L(\boldsymbol{\lambda})$ is expresses as $\delta_i^t(\boldsymbol{\lambda}) = \bar{\boldsymbol{y}}_j^t - \bar{\boldsymbol{x}}_i^t \quad for\, i = j$.

Cutting plane algorithm: we use a subgradient algorithm enhanced with valid cuts and a dual heuristic. The subgradient algorithm starts by fixing the value of the Lagrangian variables k and solving for the primal variable vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Then the Lagrangian variables are updated based on the violation of the relaxed constraints.

Variable neighborhood search: We employ a variable neighborhood search based on transformations of solutions that determine one neighborhood structure on the solution space [7]. The structure uses a finite set of pre-selected neighborhood structures denoted by Ξ_ω. We start the algorithm with 1) initialization that selects the set of neighborhood structures Ξ_ω, for $\omega = 1, \ldots, \omega_{max}$ used in the shaking phase, the set of neighborhood structures Ξ_v for $v = 1, \ldots, v_{max}$ used in the local search, and a stopping condition. In the 2) shaking step, the incumbent solution is perturbed. The algorithm generates a solution $\tilde{\boldsymbol{x}}'$ at random from ω^{th} neighborhood Ξ_ω of $\tilde{\boldsymbol{x}} = (\tilde{x}_1, ..., \tilde{x}_i)$. It

takes sensor location to be inserted at random, if it satisfies $DelaySavings(\tilde{\boldsymbol{x}}') > DelaySavings(\tilde{\boldsymbol{x}})$ and find location to be deleted at random. In the 3) local search step, the algorithm explores neighborhood to find the best neighbor $\tilde{\boldsymbol{x}}''$ of $\tilde{\boldsymbol{x}}'$ in $\Xi_v(\tilde{\boldsymbol{x}}')$. In the 4) move or not step, if the local optimum $\tilde{\boldsymbol{x}}''$ is better than the incumbent, move there ($\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{x}}''$), and continue the search with $\Xi_1(\omega \leftarrow 1)$; otherwise, set $\omega \leftarrow \omega + 1$.

4 Conclusion

Offering an efficient solution to urban traffic congestion, this paper explores the dynamic relocation of sensors to improve the network delay by controlling traffic signals under demand uncertainty. The proposed methodology to CV technology with micro simulation can be applied to any sensor location problem handled with portable devices. Lagrangian relaxation and the cutting plane method add a valid cut with a better bound, and the second subproblem is solved faster with a variable neighborhood search method. Among three multi-period stochastic models, the look-ahead policy provides the maximum benefit, and accelerated diminishing delays with additional sensors. With limited budget, the traffic operation may achieve maximum benefit by having more relocations. The proposed model can be applied repeatedly in each stage in a rolling horizon.

References

[1] X. Zhou and G,F, List, "An Information-Theoretic Sensor Location Model for Traffic Origin-Destination Demand Estimation Applications", Transportation Science 44 (2), 254-273, 2010.

[2] P.B. Mirchandani and M. Gentili and Y. He "Location of vehicle identification sensors to monitor travel-time performance", in IET Intelligent Transport Systems, 3 (3), 289-303, 2009.

[3] H. Park and A. Haghani, "Optimal Number and Location of Bluetooth Sensors Considering Stochastic Travel Time Prediction", Transportation Research Part C: Emerging Technologies 55, 203-216, 2015.

[4] Y.J. Chow and H.R. Sayarshad, "Reference Policies for Non-myopic Sequential Network Design and Timing Problems" Networks and Spatial Economics, 16 (4), 1183-1209, 2016.

[5] A.P. Scott and K. Evangelos, "Critical Intersection Signal Optimization During Urban Evacuation Utilizing Dynamic Programming", Journal of Transportation Safety & Security, 3 (1), 59-76, 2011

[6] H. Park, A. Haghani, S. Gao, S. Samuel, M.A. Knodler, "Anticipatory sensor location problem in connected vehicle environment", Transportation Science, Forthcoming

[7] N. Mladenović and P. Hansen, "Variable neighborhood search", Comp. Oper. Res., 24 (11), 1097-1100, 1997

Trade-offs in shared transportation services

Margaretha Gansterer
Department of Business Decisions and Analytics
University of Vienna
Email: margaretha.gansterer@univie.ac.at

Richard F. Hartl
Department of Business Decisions and Analytics
University of Vienna

Sarah Wieser
Department of Business Decisions and Analytics
University of Vienna

1 Introduction

Competitive markets, increased fuel costs, underutilized vehicle fleets and stricter customer demands are characteristics that currently define the logistics sector. Due to the increasing competitive pressure, many transport companies have optimized their operations up to an extent where further improvements are not achievable on an individual level [1]. A study conducted by IFEU [2] observed that on average trucks on European roads are only half-full, where nearly a quarter of these trucks run empty.

The implementation of collaboration networks is an approach that could help tackle this growing lack of efficiency. Past studies have demonstrated that collaboration among competitors can result in considerable cost savings [3]. In addition, these horizontal alliances have been linked to various environmental benefits, including the reduction of CO_2 emissions, road congestion and noise pollution. Due to this immense potential, freight-sharing has recently become a widely studied subject in the field of vehicle routing. In reality, however, transport companies have been reluctant to enter horizontal collaborations. Potential participants of collaboration networks have expressed concerns on working with competitors. They fear that instead of profiting from synergy effects, they will lose valued customers and give up potentially damaging information to their competition. A fair workload and profit distribution is considered to be one of the most important

aspects to enable horizontal collaborations in real-world applications.

In this context, we suggest that certain constraints can be imposed to set up acceptable freight-sharing frameworks among carriers. These constraints relate to (i) specific sets of customer carriers do not want to share, (ii) minimum number of customers, and (iii) minimum post-collaboration profits achieved by the carriers. We observe that these additional restrictions for single vehicle problems eliminate possible benefits. However, for the more realistic multi-vehicle cases, they come for a very low cost.

2 Problem formulation and solution approach

The underlying problem is a multi-depot pickup and delivery problem as an extension of the classical pickup and delivery problem. Each depot belongs to and therefore represents one carrier. Each carrier is associated with a number of paired requests consisting of a pickup and delivery point, which will be referred to as the initial customer distribution. Additionally, each carrier is equipped with a certain number of vehicles, starting and returning to their depot. Each vehicle is only capable of performing one tour. Due to the context of the problem, a delivery point has to be served after its associated pickup point by the same vehicle. Both a single and a multi-vehicle case are examined in our study. In the Multi-Depot TSP with Pickups and Deliveries (MDTSPPD) each carrier only has access to one vehicle with unlimited capacity. The problem is extended to a multi-vehicle case (MDVRPPD), including duration and capacity constraints for each vehicle.

Solutions of MDTSPPD as well as MDVRPPD may lead to unevenly distributed solutions where all customers are assigned to only one carrier. In a collaborative setting, this is clearly not desirable and will scare-off potential participants. Companies may be more likely to enter collaborations if they can, e.g, keep some of their current customers. This is a reasonable request, given that many companies have valuable long-term customers that they do not want to lose [4].

We examine the potential of collaborative solutions by considering three different fairness constraints for MDVRPPD. First, the requirement that each carrier will get to keep a minimum amount of their initial customers (A). Second, the effect of only keeping a certain number of customers - regardless of their initial carrier - will be examined (B). This way an upper bound on customer share loss can be established. Finally, a minimum profit with respect to the status quo is set, resulting in an upper bound on profit losses (C).

While relatively small instances can be solved to optimality, solutions for larger instances are generated using an adaptive large neighborhood search (ALNS) algorithm. The ALNS was chosen because it is widely used in the field of PDP problems and has been proven to find good solutions in a reasonable time. It is applied to five cases. First, the costs of a non-collaborative situation constrained by the initial customer distribution are computed. This means each carrier faces a

Table 1: The cost of keeping $\frac{1}{3}$ or $\frac{2}{3}$ of the initial customers. We report the decrease in collaboration profit compared to the total collaboration profit without constraints A-C (No).

	MDTSPPD			MDVRPPD		
Instances	No	$\frac{1}{3}$ kept	$\frac{2}{3}$ kept	No	$\frac{1}{3}$ kept	$\frac{2}{3}$ kept
O1	12.53%	-6.16%	-9.16%	8.46%	-0.86%	-1.64%
O2	24.28%	-9.21%	-16.34%	18.12%	-1.76%	-5.05%
O3	39.38%	-21.03%	-31.46%	25.22%	-2.95%	-11.33%

classical PDP with only one depot and one (MDTSPPD) or multiple (MDVRPPD) vehicles at their disposal. Additionally, the collaborative solutions are determined with and without constraints (A-C). The ALNS is based on the work of [5]. However, two problem-specific operators are introduced. These operators take constraint violations (A-C) into account.

3 Computational study

The computational experiment aims to quantitatively measure potential benefits of collaborative solutions in comparison to the status quo, as well as the trade-offs when constraints (A-C) are introduced. We use an extensive set of test instances covering different problem characteristics. In particular, different degrees of customer area overlaps are considered [6]. These scenarios are denoted as O1 (low overlap), O2 (medium overlap), and O3 (high overlap).

We present numerical results, where we observe that freight-sharing among carriers can lead to cost savings of around 25% on average. These findings go in line with past studies, where collaborative gains of 20-30% are reported [3]. We show that the cost savings could even go up to around 35-40% when there is a strong regional overlap of customers. Despite these large potential savings, companies hesitate to enter horizontal collaborations for the fear of customer and profit loss. Our study supports this obstacle by showing that for some instances, one carrier ends up serving nearly all customers. Additionally, in almost all instances customers are unevenly distributed among carriers. This can be explained by the fact that once a customer from a competitor is included in a tour, all other customers of that carrier can easily be reached as well. In either case, it is clear that this distribution is not desirable for potential participants. These findings enforce the need for constraints A-C to generate acceptable solutions for all participants. The effect of introducing these constraints is quantified and analyzed.

In Table 1 results for constraint type A are presented.

The results show that constraint A is detrimental in case of MDTSPPD problems: given a potential collaboration profit of 39.38% of the initial solution of instances O3, constraints of type A decrease this by 31.46%. Leaving less than 8% post-collaboration gain. However, this does not

hold for MDVRPPD settings. Our results show that for these - more realistic settings - constraints of type A can be imposed by relatively low cost. On average, the collaboration profit decreases by about 6%. We can show that similar results can be obtained for constraints B and C. Constraint B comes with particularly low cost. On average, less than 3% are lost compared to a solution where no constraints are introduced.

4 Conclusion

The aim of this study is to assess potential trade-offs in collaborative pickup and delivery problems under a centralized collaboration framework. Trade-offs relate to the fact that carriers do not want to share their full set of customers with collaboration partners. We observe that the cost of introducing such constraints rises with the degree of regional customer overlap. Overall the study demonstrates that collaborations can provide a high potential for cost savings even if these constraints are introduced. Carriers may therefore explicitly exclude certain long-term valued customers from being shared and still benefit from the collaboration.

References

[1] C. Vanovermeire, K. Sörensen, A. Van Breedam, B. Vannieuwenhuyse and S. Verstrepen, "Horizontal logistics collaboration: decreasing costs through flexibility and an adequate cost allocation strategy", *International Journal of Logistics Research and Applications* 17(4), 339-355 (2014).

[2] IFEU (2008), EcoTransIT: Ecological Transport Information Tool. Tech. rep. Institut fr Energie- und Umweltforschung Heidelberg, 2008.

[3] M. Gansterer and R.F. Hartl, "Collaborative vehicle routing: a survey", *European Journal of Operational Research* 268, 1-12 (2018).

[4] M. Gansterer, R.F. Hartl and S. Wieser "The cost of continuity in the collaborative pickup and delivery problem", *Lecture Notes in Computer Science* 11184, 239-252 (2018).

[5] S. Ropke and D. Pisinger "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows", *Transportation Science* 40.4, 455-472 (2006).

[6] S. Berger and C. Bierwirth "Solutions to the request reassignment problem in collaborative carrier networks", *Transportation Research Part E: Logistics and Transportation Review* 46.5, 627-638 (2010).

The co-development of railway and land use in Sydney

B.Lahoorpoor
Department of Civil Engineering
University of Sydney
Email: bahman.lahoorpoor@sydney.edu.au

D.Levinson
Department of Civil Engineering
University of Sydney
Email: david.levinson@sydney.edu.au

1 Introduction

It has been observed that there is a strong relationship between land development and infrastructure investments. Transforms in land use change the travel demand patterns and in turn, the altered traffic flows on transportation networks drives the investment on transportation facilities. This development leads to changing the accessibility pattern, which drives the re-location of activities and land uses. Figure 1 illustrates the schematic co-evolving relationship between transportation networks and urban transformation. During this feedback loop, both transport networks and land use are continuously evolving, leading to urban spatial transformation. It is worthwhile to mention that other exogenous factors such as new technologies, traffic management, policy on economic growth, land availability, spatial policies, etc. play a role on this process.

There are only a few number of study that take the long-run full cycle of land use and transport networks interaction into account and only a few cases examine population changes in urban areas [1, 2, 3, 4, 5, 6]. In part, that is due to the lack of more sophisticated analytical techniques to analyze network elements, in part due to historical data availability.

In most analysis, the relationship between infrastructure and travel demand has been considered as one-way direction in which the infrastructure network (supply) as the explanatory variable and traffic (demand) as the dependant variable [1]. The reasons of increases in traffic can be categorized in the temporal changes: shifts in route, mode, or departure time in the short term; changes in

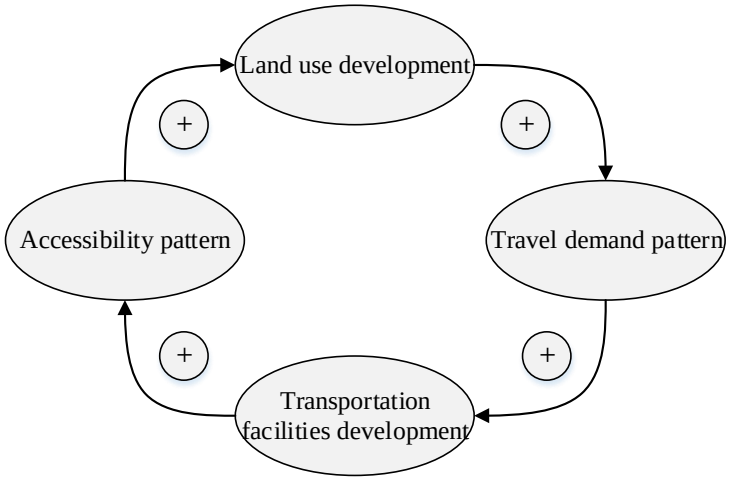


Figure 1: Land use and transit network interaction

destination and more trips by trip-makers in the medium term; and more trip-makers in the long run [1].

While these studies provide better understanding of the characteristics of transit networks, there is a lack of knowledge on how transit networks and land use could evolve into their current unique state, form, and structure patterns as they are born, grow, mature, and decline over time. Although a variety of actors are involved in developing an urban transit networks that pursue their interest [4], there is a research gap to understand the co-evolution of land use and large-scale transit systems, particularly railways networks embedded in a metropolitan area.

This article considers the relationship between transit infrastructure and land development, and examines the railway network as a centralized/decentralized force. It is widely believed that high population density is an important factor in the success of rail systems (density represents potential ridership). However, just because rail depends on high population density for success does not inherently mean that either high density areas generate rail investment or rail creates high-density areas around stations [1].

In this article, it is tested whether high density land development encourages the investment in rail infrastructure and in return rail infrastructure increases densities. Similar to all other transit networks, the railway enables movement, and as such, it increases the densities for certain activities in some places and for other activities in different places. By increasing densities for jobs in CBD, it is simultaneously decreasing housing densities in those places by making housing in the core more expensive and making housing outside the core have greater accessibility. This joint process of infrastructure and land development location is called co-development which transport drives land use, land use derives transport network [1, 7]. This research begins a longer investigation.

Sydney, as a great example of a rapidly developing city, had public railway transport services beginning in the 1850s, which facilitated and responded to the development of suburbs. The

advent of first steam railways occurred in 1855 which formed the basis of the New South Wales Government Railways. The first line was opened for passenger and freight trains between Sydney and Granville, which at the time was a center of agriculture. Railways were soon complemented by an extensive tram system, but the transit system was in the twentieth century disrupted by the rise of the automobile. Figure 2 demonstrates the evolution of Sydney railway network during the time.

Figure 2: Evolution of Sydney Railway Network

Railway service, by increasing the travelled distance in a given time over previous transport modes enabled commutes to be lengthier and thus made more area accessible for residences at a given commute time [1]. As a result, population moves out from the core of a city to the outer suburbs in specific commute times which increase the population density in suburbs. In order to investigate the effect of railway on land use and how land use changes a railway network the following hypothesis are tested based on the available historical data:

- Population density
  - Population density in the periphery is positively associated with the lagged increase in density of new rail stations

  - population density in the periphery is positively associated with the lagged population density of the nearest suburbs (neighbor effect)
  - Population density in the periphery is positively associated with the lagged network density of the nearest suburbs (neighbor effect)
  - Population density in the core is negatively associated with the lagged increase of new rail stations.

- Network density
  - Network density in the periphery is positively associated with the lagged increase in population density
  - Network density in the periphery is positively associated with the population density of the nearest suburbs (neighbor effect)
  - Network density in the periphery is positively associated with the network density of the nearest suburbs (neighbor effect)
  - Network density in the core is negatively associated with the lagged increase in population density

References

[1] D. Levinson, "Density and dispersion: the co-development of land use and rail in london," *Journal of Economic Geography*, vol. 8, no. 1, pp. 55–77, 2007.

[2] F. Xie and D. Levinson, "Topological evolution of surface transportation networks," *Computers, Environment and Urban Systems*, vol. 33, no. 3, pp. 211–223, 2009.

[3] D. M. Levinson, F. Xie, and S. Zhu, "The co-evolution of land use and road networks," 2007.

[4] O. Cats, "Topological evolution of a metropolitan rail transport network: The case of stockholm," *Journal of Transport Geography*, vol. 62, pp. 172–183, 2017.

[5] D. M. Levinson, D. Giacomin, and A. Badsey-Ellis, "Accessibility and the choice of network investments in the london underground," *Journal of Transport and Land Use*, vol. 9, no. 1, pp. 131–150, 2016.

[6] D. Kasraian, K. Maat, D. Stead, and B. van Wee, "Long-term impacts of transport infrastructure networks on land-use change: an international review of empirical studies," *Transport Reviews*, vol. 36, no. 6, pp. 772–792, 2016.

[7] D. Levinson and F. Xie, "Does first last? the existence and extent of first mover advantages on spatial networks," *Journal of Transport and Land Use*, vol. 4, no. 2, pp. 47–69, 2011.

Dynamic traffic assignment for multimodal GSOM models

Megan M. Khoshyaran

ETC Economics Traffic Clinic, 34 av. des Champs-Elysées,

F75008 Paris, FRANCE

Jean-Patrick Lebacque

UPE/IFSTTAR, COSYS/GRETTIA, 14-20 Bd Newton,

F77447 Marne-la-Vallée, FRANCE

E-mail: jean-patrick.lebacque@ifsttar.fr

1 Introduction

Transportation systems are evolving fast under the impact of many factors. These include car manufacturers, who are actively promoting communicating vehicles, and vehicles exhibiting various levels of automation or autonomy. They are also proposing new technology for motorization (electric engines) and new services such as on demand vehicles. Other factors result from economic pressure and environmental concerns. Social networks also contribute significantly: they offer new services such as Uber, or Waze. The latter also contributes to the propagation of information through the network. Travellers are more and more connected: this opens the door to effective seamless multimodality. Even the first/last mile transportation may now be taken in charge by a micro-mobiliy based on new individual electrical vehicles and implements. Regional agencies are therefore concerned and need fully multi-modal models that have the capacity to describe the new complex transportation systems, including communication and information flows. For instance information impacts of course routing [1], [2] but also may have significant large scale impact [3]. Regional agencies also need tools to estimate regional equilibria, both for long term planning purposes and short term network management. Such concerns have been considered for instance in [4], [6], [7] and [5] . The study introduced in this paper aims to contribute to addressing these issues.

2 The model

The multimodal model proposed in this paper is based on the GSOM model introduced in [8], [9], and applied to networks with information in [10], but in the context of reactive dynamic assignment. The model was extended to fully multmodal transportation systems in [11] and [12]. The main features of this extension are the following:

- There are two flows, vehicular and passengers. The passenger flow is subordinated to the vehicular flow (vehicles carry passengers).
- Vehicles end passengers may be endowed with attributes which are passive and neutral (such as direction and path), passive (passenger or vehicle type and caracteristics), or active (information, battery charge, engine temperature etc).
- vehicular dynamics follow regular first order-like dynamics. Passengers are described as a specific attribute of vehicles, the passenger load. Passive attributes are advected, active attribute dynamics follow an advection equation with source terms.

Thus the basic notations of the model are the following:

- x the position, t the time;
- $\rho(x,t)$ the density, $v(x,t)$ the speed and $q(x,t)$ the flow of vehicles.
- $I(x,t)$: the vector of attributes. It comprises:
  - $\varpi(x,t)$: the load of passengers;
  - some neutral passive attributes required for assignment, notably $\chi(x,t) \stackrel{def}{=} \left(\chi^d(x,t)\right)_{d\in\mathcal{D}}$: the vector of fractions of vehicles with destination $d \in \mathcal{D}$ ($\chi^d(x,t)$ denotes the fraction of vehicles at time t and location x with destination d) and $\mu(x,t) \stackrel{def}{=} \left(\mu^d(x,t)\right)_{d\in\mathcal{D}}$: the vector of fractions of passengers with destination $d \in \mathcal{D}$ ($\mu^d(x,t)$ denotes the fraction of passengers at time t and location x with destination d);
  - $\kappa(x,t)$ denotes a vector of supplementary attributes pertaining to passengers or vehicles, possibly active.
- ϖ has the unit of number of passengers per vehicles, thus $\rho\varpi$ denotes the density of passengers per unit length of links. The speed of passengers is equal to the speed of vehicles $v(x,t)$. It follows that the flow of passengers is given by:

$$p \stackrel{def}{=} \rho\varpi v \tag{1}$$

- Velocity and density are connected through the fundamental diagram:

$$v = V_e(\rho,\kappa) \tag{2}$$

- The vehicles have an attribute which is their capacity with respect transport passengers, ϖ_{max}. This attribute is connected to vehicles, thus must satisfy an advection equation

$$\partial_t \varpi_{max} + v \partial_x \varpi_{max} = 0 \tag{3}$$

The passenger load is bounded by the vehicular capacity:

$$0 \leq \varpi(x,t) \leq \varpi_{max}(x,t) \quad \forall x,t \tag{4}$$

This attribute ϖ_{max} must be included into I:

$$I = (\chi, \varpi_{max}, \varpi, \mu, \kappa) \tag{5}$$

Finally the model is described by the following equations in eulerian coordinates (x,t):

$$\left[\begin{array}{ll} \partial_t \rho + \partial_x (\rho v) = 0 & (6.1) \\ \partial_t (\rho I) + \partial_x (\rho I v) = -\rho \Phi(I,\rho) & (6.2) \\ v = V_e(\rho,) & (6.3) \\ I = (\chi, \varpi_{max}, \varpi, \mu, \kappa) & (6.4) \\ 0 \leq \varpi \leq \varpi_{max} & (6.5) \end{array}\right. \tag{6}$$

The source term Φ concerns only κ. Thus passengers can board or unboard only at nodes, at which locations passengers and vehicles can change their path.

The main complexity of the transportation dynamics is recaptured at nodes, which may represent intersections, but also stations or multimodal poles. Links are monomodal, and walking occurs in nodes; passengers possibly change mode in nodes. Some of these issues, as well as some discretization issues, are addressed in [11] and [12].

3 Assignment

For dynamic assignment we will consider two main ideas

- a scheme based on cross-entropy which is a local scheme in the sense that it is arc based: path choice is carried out at each node. This scheme follows ideas outlined in [13] and [14]. It can also be related to ideas expressed in [15];
- a scheme based on a global path base fixed point approach inspired by ideas from [16].

A simple test case is shown in figure 1. Note the central node which acts as a multimodal pole. Travellers may change mode there. An example of multimodal path assignment, via fixed point search, is shown below on figure 2. Path costs and flows are depicted, with convergence shown despite the fact that the optimal solution entails mode changes. These mode changes result in path flow time-discontinuities.

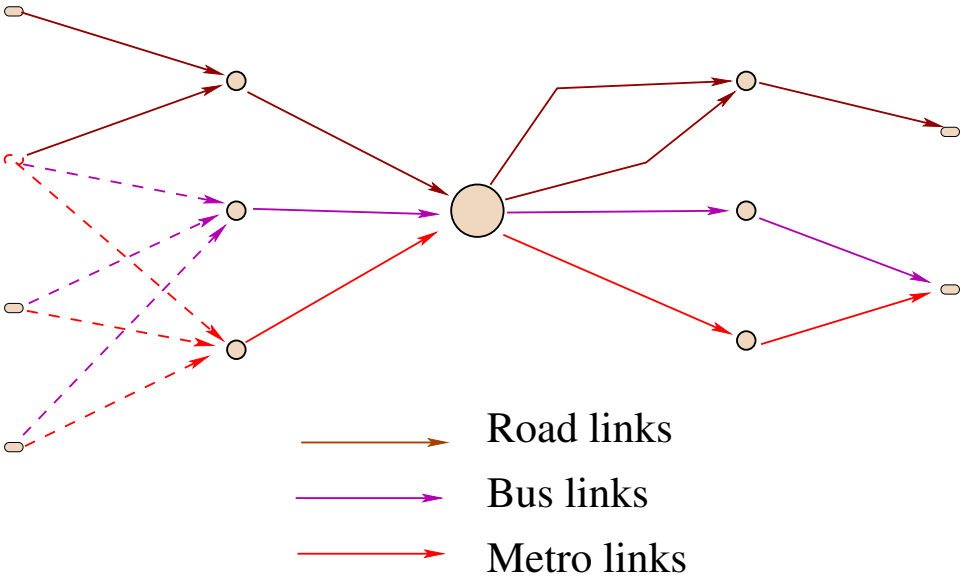


Figure 1: A simple multimodal network with road, bus, metro

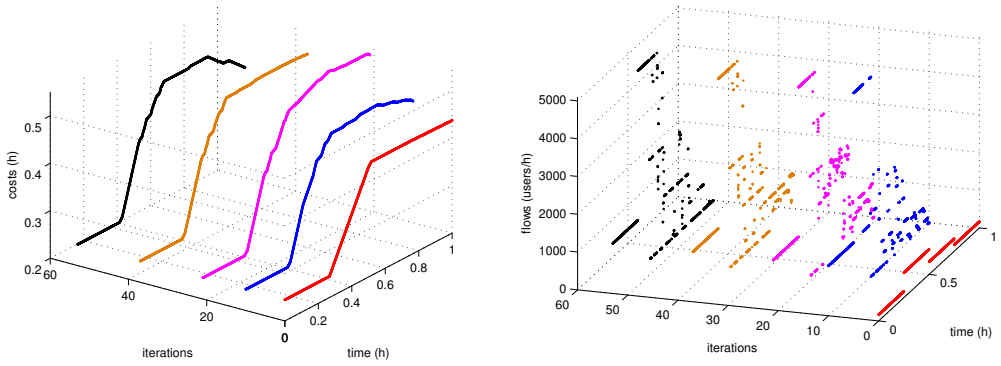


Figure 2: Optimal assignment, example of path costs (left) and path flows (right) versus time, as a function of iteration

References

[1] L. Codeca, R. Frank and T. Engel. Traffic routing in urban environments: The impact of partial information. In *Proceedings of 17th IEEE International Conference on Intelligent Transportation Systems (ITSC 2014)*, pp. 2511-2516, 2014.

[2] Medetov S, Bakhouya M, Gaber J, Zinedine K, Wack M, Lorenz P. 2014. A decentralized approach for information dissemination in Vehicular Ad hoc Networks. *Journal of Network and Computer Applications* 46,154165. 2014.

[3] Amin-Naseri, Mostafa, Pranamesh Chakraborty, Anuj Sharma, Stephen B. Gilbert, Mingyi Hong. "Evaluating the reliability, coverage, and added value of crowdsourced traffic incident reports from Waze." No. 18-03519. 2018. Transportation Research Board 97th Annual Meeting, Washington, DC, January 7-11, 2018.

[4] Di Gangi, Massimo, and Antonio Polimeni. "A model to simulate multimodality in a mesoscopic dynamic network loading framework." Journal of Advanced Transportation 2017 (2017).

[5] Meng, M., Shao, C., Wong, Y. D., and Zhang, J. (2014). A multiclass, multimodal dynamic traffic assignment model with departure time. Mathematical Problems in Engineering, 2014.

[6] Verbas, I. Ömer, Hani S. Mahmassani, and Michael F. Hyland. "Dynamic Assignment-Simulation Methodology for Multimodal Urban Transit Networks." Transportation Research Record: Journal of the Transportation Research Board 2498 (2015): 64-74.

[7] Wang, Y., Szeto, W. Y., Han, K., and Friesz, T. L. (2018). Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications. Transportation Research Part B: Methodological.

[8] J.P. Lebacque, S. Mammar and H. Haj-Salem. Generic second order traffic flow modeling. In R.E. Allsop, M.G.H. Bell and B.G. Heydecker, editors, *Transportation and Traffic Flow Theory 2007*, 2007.

[9] J.P. Lebacque and M.M. Khoshyaran. A variational formulation for higher order macroscopic traffic flow models of the GSOM family. *Transportation Research Part B*, 57:245–265, 2013

[10] Khoshyaran, Megan M., and Jean-Patrick Lebacque. "GSOM Traffic Flow Models for Networks with Information." In: International Conference on Systems Science. pp 210-220. Springer, Cham, 2016.

[11] Lebacque, Jean-Patrick, and Megan M. Khoshyaran. "Multimodal Transportation Network Modeling Based on the Generic Second Order Modeling Approach." Transportation Research Record (2018): 0361198118797486.

[12] Lebacque, Jean-Patrick, and Megan M. Khoshyaran. "Semi-lagrangian formulation of an extended GSOM Model for Multimodal Transportation Systems." IFAC-PapersOnLine 51.9 (2018): 1-6.

[13] Ma, Tai-Yu, and Jean-Patrick Lebacque. "A cross entropy based multi-agent approach to traffic assignment problems." Traffic and Granular Flow07. Springer, Berlin, Heidelberg, 2009. 161-170.

[14] Ma, T. Y., and Lebacque, J. P. (2013). A cross entropy based multiagent approach for multiclass activity chain modeling and simulation. Transportation Research Part C: Emerging Technologies, 28, 116-129.

[15] Jin, W. L. (2007). A dynamical system model of the traffic assignment problem. Transportation Research Part B: Methodological, 41(1), 32-48.

[16] Askoura, Y. Private communication. 2010

The Vehicle Routing Problem with Digital Lockers Terminals

Simona Mancini
Department of Mathematics and Computer Science
University of Cagliari, Italy
Email: simona.mancini@polito.it

1 Introduction and motivation

In the last decade, the advent of e-commerce radically changed the shopping habits. Nowadays, customers can compare, in very few minutes, a huge number of alternatives and offers, directly from their laptop, tablet, smartphone or even smartwatch, without leaving their house or their office. Home delivery has established new standards in terms of quality of service, and the number of users choosing to adopt this purchasing method is constantly growing. The large increment of home delivery requests started to have a crucial impact in last mile delivery, as pointed out in [2]. In fact, given the large amount of request, companies cannot perform the delivery in the moment preferred by the customers (generally at the end of the day when they are at home) and it is obliged to increase the length of the delivery window to a buffer of several hours, within which the customer is asked to be home if it does not want to miss the delivery. This will results in a negative impact on the quality of service perception by the user and, consequently, on the customer satisfaction. Moreover, this issue generated drawbacks not only for customers but also for delivery companies which are often obliged to perform twice the delivery, because at the first attempt the customer was absent. This would create a decrease of efficiency in the logistic companies, in terms of costs, and an increment of traffic congestion in urban areas. To overcome this issue, in the very last years a new delivery system, named unattended delivery, in which delivery are performed to shared facilities, such as Digital Lockers Terminals (DLTs), has been introduced. These facilities are generally located into a supermarket open 24h/day, at a train station, or in other places with a very wide opening window. The advantage of such a system is twofold. Customers do not have to attend the delivery at home but can autonomously pick-up their goods when it is more convenient for them. On the other hand, transport companies may perform the delivery at anytime and can consolidate goods destined to different customers but associated to the same DLT, reducing the

number of locations to visit, with a positive impact on both delivery costs and traffic congestion. An analysis of the economic benefits of such a system have been provided in [2] and [1] and , while an analysis of the reaction of customers to this new trend in parcel delivery has been reported in [3]. Despite the evident advantages of a DLT based distribution system, this strategy still has some drawbacks. In fact, in rural areas, where the diffusion of DLT is still very limited, customers must cover several kilometers to pick-up their goods and their level of satisfaction may sensibly decrease. Furthermore, old aged people or customers with disabilities may experience some difficulties to reach the DLT, even if it is not far from their house. Logistic companies, such Amazon, offer a service in which each customer can chose between two delivery alternatives. Delivery can be performed at home (or at another place indicated by the customer, such as the office) without any indication about the moment of the day in which it will be carried out (therefore the customer must stay at home waiting for the delivery) or the delivery can be performed at the DLT indicated by the customer when he can pick it up when it is more convenient for him. The aim of this paper is to propose a new delivery system which combine home delivery with DLT deliveries in a more convenient way in order to increase both customers satisfaction and companies revenue. In this newly proposed system, customers may choose between three options for the delivery:

1. to receive their delivery at home within a short time window they indicate within which they must attend at home
2. to receive their delivery at one of the DLTs they indicated in their order (one can choose DLTs near his house, near the office, the gym where he goes in the evening, the house of his parents, and so on..) receiving a small compensation for the discomfort to pick it up at the DLT
3. to let the company to decide whether to delivery their package at home, within their preferred time window or in one of the DLTs they indicated, obtaining a small compensation

In this system, people who need to receive home delivery will choose option 1, people who are very interested into receiving the compensation, or who, for personal reasons, prefer not to receive their delivery at home, will choose option 2, while for customers for which it is indifferent where to receive their delivery the company can choose the most convenient option in their delivery planning. This way, customers satisfaction increases for each category of customers and the company may reduce its transportation costs without downgrading the quality of service.

2 Mixed Integer Programming Formulation

The goal of the problem is to serve a set of delivery requests I starting from a depot, 0. Each delivery must be performed at customer location or at one of the DLTs indicated by the customers. A set F of DLTs is available, but each request i is compatible only with a subset of F. To each DLT, f, is associated a maximum number of request that can be contemporaneously assigned to it,

B_f, representing the number of empty lockers at f. A service time s_i is defined for each customer and DLT. The service time associated with a DLT does not depend on the number of packages delivered to it. Each customer i can be served only within a fixed time window $[E_i, L_i]$ while DLTs can be accessed at any time. Let us define the set $N = I \cup F$ and $N_0 = N \cup 0$. For each pair of nodes i, j in N_0 are known travel time, t_{ij} and travel cost, c_{ij}. Each vehicle start at the depot and must return to the depot within a given time limit T_{max}. We indicate with δ the compensation paid to a customer if its delivery has been performed to a DLT, while with γ the fixed cost related to the usage of each vehicle. The objective is to minimize total distribution costs, given by the sum of travel costs, vehicle usage costs and compensations paid to the customers. Without loss of generality we assume that each DLT may be visited by at most one vehicle. In fact, given the small size of objects that can be delivered to a DLT, respect to the capacity of the vehicles, and the small number of available lockers, we can assume that the demand of a DLT can be fulfilled by a single vehicle. Therefore, in an optimal solution it will never happen that a DLT will be served by more than one vehicle.Before to report the mathematical formulation we need to introduce the following decision variables types:

X_{ij}: binary variables representing whether node i is visited just after node j or not

Y_{if}: binary variables representing whether customer i order is delivered to DLT f or not

Z_f: binary variables representing whether DLT f is visited or not

T_i: non-negative variables representing the time in which customer i is reached

The formulation is reported in the following:

$$\min \sum_{i \in N_0} \sum_{j \in N_0} c_{ij} X_{ij} + \delta \sum_{i \in I} \sum_{f \in F} Y_{if} + \gamma \sum_{j \in N} X_{0n} \tag{1}$$

$$\sum_{i \in N_0} X_{ij} + \sum_{f \in F} Y_{jf} = 1 \quad \forall j \in I \tag{2}$$

$$\sum_{i \in N_0} X_{ij} = \sum_{i \in N_0} X_{ji} \quad \forall j \in N_0 \tag{3}$$

$$Z_f \geq \frac{1}{|I|} \sum_{i \in I} Y_{if} \quad \forall f \in F \tag{4}$$

$$\sum_{i \in N_0} X_{if} = Z_f \quad \forall f \in F \tag{5}$$

$$T_j \geq T_i + t_{ij} + s_j - 2T_{max}(1 - X_{ij}) \quad \forall j \in N \ \ \forall i \in N_0 \tag{6}$$

$$-T_{max} \sum_{f \in F} Y_{if} + E_i \leq T_i \leq +L_i + T_{max} \sum_{f \in F} Y_{if} \quad \forall i \in I \tag{7}$$

$$T_j + s_j + t_{j0} \leq T_{max} \quad \forall j \in N \tag{8}$$

$$\sum_{i \in I} Y_{if} \leq B_f \quad \forall f \in F \tag{9}$$

The objective function is reported in 1. Constraints 2 imply that each order must be delivered or directly to the customer or to one of the compatible $DLTs$. Constraints 3 ensure route continuity. If at least one order has been assigned to a DLT, it must be visited, as specified by the combinations of Constraints 4 and 5. The arrival travel time at each node is ruled by Constraints 6. Customers time windows must be respected if and only if its order is delivered directly at customer's location, as stated in Constraints 7. Each vehicl must return to the depot before T_{max} as implied by Constraints 8.Finally, the number of orders delivered to a DLT must not exceed its capacity, as imposed by Constraints 9.

3 A Matheuristic for the VRP-DLT

To solve large size instances a matheuristic approach is proposed. Starting from an initial feasible solution, at each iteration, p customers are randomly draw. All the other $N-p$ customers are forced to be assigned to the same DLT they were assigned in the current solution, or forced to be directly served if they were directly served in the current solution. The selected p are let free to be assigned to a DLT or directly served. This overconstrained version of the model is solved with a very short time limit and the best feasible solution is kept as current solution.

4 Computational Results

Computational results, carried out on instances of different size, show the efficiency and effectiveness of the proposed matheuristic approach. A study on the impact of an increment of the number of DLTs on the total delivery cost has been performed. Furthermore, a comparison of this mixed delivery strategy with the classical strategy in which all the customers are directly served at home, and the strategy in which all the customers are served through DLTs, is performed and the obtained results show that the mixed strategy is the most convenient both for the transport companies and for the customers. All detailed results, with a deep analysis and discussion, will be presented at the conference.

References

[1] S.Iwan, K. Kijewska and J. Lemke. "Analysis of parcel lockers efficiency as the last mile delivery solution the results of the research in Poland", *Transportation Research Procedia* 12, 644-655 (2016).

[2] E. Morganti, S. Seidel, C. Blanquart, L. Dablanc and B. Lenz, "The impact of e-commerce on final deliveries: alternative parcel delivery services in France and Germany ", *Transportation Research Procedia* 4, 178-190 (2014).

[3] Y. Vakulenko, D. Hellström and K.Hjort, "What's in the parcel locker? Exploring customer value in e-commerce last mile delivery", *Journal of Business Research* 88, 421-427, (2018)

Passenger-to-Itinerary Assignment Model for Congested Urban Rail Networks

Yiwen Zhu
Microsoft Corp.

Haris N. Koutsopoulos
Department of Civil and Environmental Engineering
Northeastern University

Nigel H.M. Wilson
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Email: zoe8200@gmail.com

1 Introduction

With increased urbanization and densification, public transport ridership has increased dramatically in many urban areas. As a result, crowding and congestion have also increased. With capacity limited, the level of service can deteriorate significantly. The adoption of means of collecting data automatically from systems such as Automatic Fare Collection (AFC) and Automatic Vehicle Location (AVL) facilitates the development of relevant metrics and the monitoring of system state without costly manual data collection (Bagchi and White, 2005; Agard et al., 2006; Zhao et al., 2007; Chan, 2007; Pelletier et al., 2011; Ortega-Tong, 2013; Langlois et al., 2016; Koutsopoulos et al., 2017).

The paper introduces a probabilistic Passenger-to-Itinerary Assignment Model (PIAM) that is applicable under capacity constraints for trips with and without transfers and route choices. PIAM infers details of the journey passengers made on a particular day based on the actual AFC and AVL data from that day, while the traditional schedule-based assignment models (e.g. Nuzzolo et al., 2001; Poon et al., 2004; Hamdouch and Lawphongpanich, 2008; Sumalee et al., 2009; Nuzzolo et al., 2012) are mainly planning tools that target future conditions.

At the disaggregate level, PIAM infers individual passenger movements at a high resolution (journey time components, passenger location inference, etc.). At the aggregate level, the output (route choice fractions, train load estimation, journey time decomposition, etc.) provides useful performance metrics for operators to assess the capacity utilization and evaluate the impact of

near capacity operations on passengers.

2 Methodology

We assume a closed AFC system, where both the tap-in and tap-out times of passengers are known. Train arrival and departure times at stations are also known from the AVL system. For a passenger with transfers, *itineraries* represent different combinations of trains to complete all the segments for this journey on the chosen route. The main challenges with the general assignment problem are: (i) the number of possible itineraries can be very large, especially for trips involving transfers, and (ii) the route choice inference. To address those problems, we assume that the probability of being left behind at a transfer station is the same for transfer and non-transfer passengers (i.e. passengers who enter the system at that station) who arrive on the boarding platform at the same time. We further use left behind probabilities estimated from passengers without route choices to infer the route choice fractions for passengers with transfers.

Figure 1 illustrates the framework of PIAM: i) the left behind model estimates the probability of being left behind using data from trips without route choice or transfers; ii) The route choice model estimates the route choice fractions by time interval given the left behind probabilities; iii) The assignment model assigns passengers to itineraries based on the left behind probabilities and route choice fractions. The dimensionality and complexity of the assignment problem is reduced, especially for transfer trips.

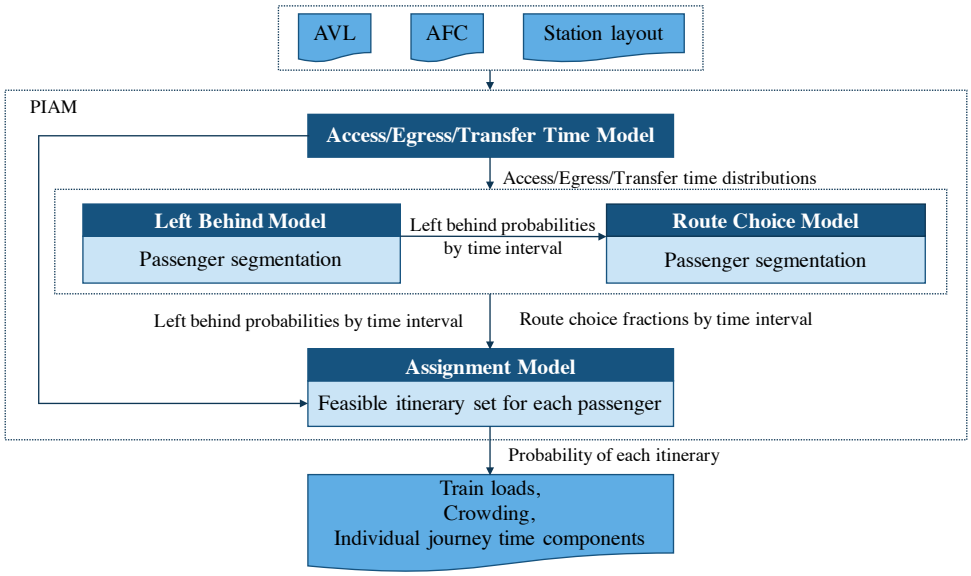


Figure 1: PIAM framework with route choice

Access/Egress/Transfer Time Model

Zhu (2017) proposed a model for the estimation of access/egress time distributions using AFC and

AVL data that consists of two components: the walk speed model, and the walk distance model. The model can be directly extended for the estimation of transfer time distribution.

Left Behind Model

An important assumption of PIAM is that the probability of being left behind is the same for transfer and non-transfer passengers at the same station and time period (based on the arrival time at the platform). The left behind probabilities, at the aggregate level, can be estimated using the approaches proposed in Zhu et al. (2017), based on data from trips without transfers. Assuming that the access/egress speed distributions are known, the likelihood of observing the journey times of all passengers in the group can be derived. Zhu et al. (2017) examined maximum likelihood and Bayesian inference methods to estimate the left behind probabilities.

Route Choice Model

Figure 2 illustrates the possible instances for a passenger with two routes (each with one transfer). $P(r_1)$ and $P(r_2)$ denote the probabilities of choosing routes r_1 and r_2 respectively. After tapping-in, the passenger walks to the boarding platform, and he/she may arrive during different trains' departures. After alighting at the transfer station, he/she may arrive during different trains' departures for the next segment. If the coming train is full, the passenger will be left behind (P_n represents the probability of left behind n times).

For a trip with multiple routes, the conditional probability of using each itinerary and tapping-out at the observed tap-out time, given the corresponding route choice, can be derived according to Figure 2, and is a function of the access/transfer/egress time distributions and the left behind probabilities. The probability of choosing different routes, i.e. $P(r)$, can be estimated by maximizing the total likelihood of the observed tap-out times for all passengers.

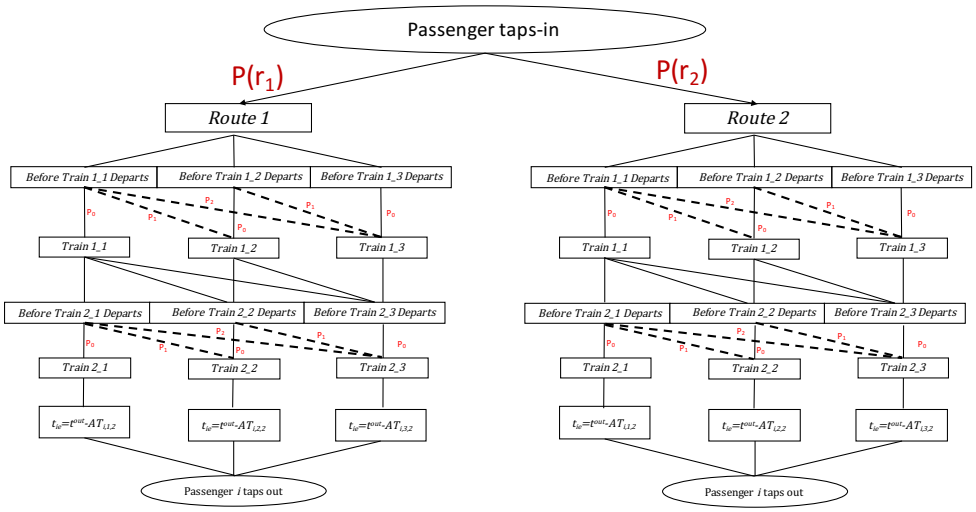


Figure 2: PIAM structure for a passenger with route choice

Assignment Model

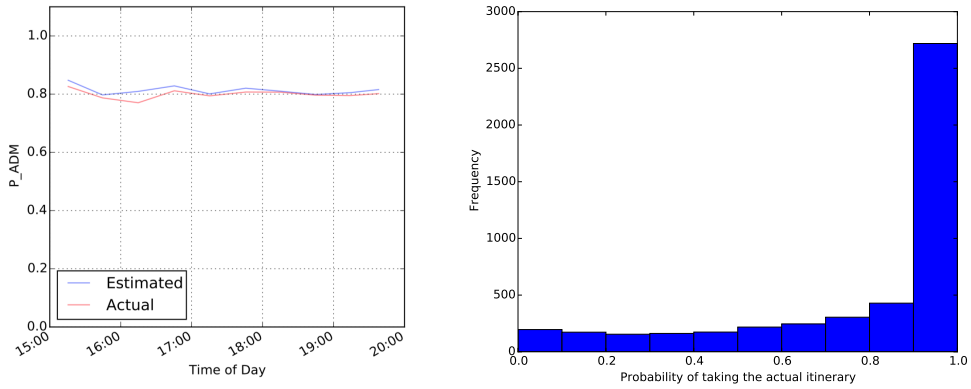
With left behind probabilities and route shares estimated by time interval, the probability of using different itineraries for a given passenger, can be calculated based on Bayesian Theorem using the graph depicted in Figure 2.

3 Model Validation

In order to validate the proposed method, synthetic data for a small portion of the network, was generated using actual tap-in times and train movement data with four OD pairs. Passengers with route choices were randomly assigned to a path according to pre-defined fractions. At all stations, transfer/tap-in passengers are loaded onto the trains based on a first come, first served (FCFS) basis according to their arrival time at the boarding platform.

The left behind probabilities at all the origin/transfer stations are estimated by station and time interval. The route choice fractions were estimated using maximum likelihood. Given the estimated route fractions and left behind probabilities, the assignment model is used to estimate the probability of each feasible itinerary for each passenger.

Figure 3a shows the estimated probability, of choosing path r_1 compared with the "true" in the synthetic data in 30 min intervals. The estimated probability is consistent with the actual (synthetic) data. Figure 3b shows the distribution of the estimated probabilities of assigning to the actual itinerary for passengers with route choices. The probabilities of assigning passengers to their actual routes and itineraries are high, despite the large number of feasible itineraries for many passengers (up to 60 in some cases). As the assignment is estimated without dependency on the upstream (which is usually the case for traditional assignment), the estimation can be run in parallel for the OD pairs of interests.



(a) Route Choice Estimation

(b) Itinerary Assignment

Figure 3: Model Validation

References

Agard, B., Morency, C., and Trépanier, M. (2006). Mining public transport user behaviour from smart card data. In *12th IFAC Symposium on Information Control Problems in Manufacturing-INCOM*, pages 17–19.

Bagchi, M. and White, P. (2005). The potential of public transport smart card data. *Transport Policy*, 12(5):464–474. Road User Charging: Theory and Practices W. Saleh.

Chan, J. (2007). Rail transit od matrix estimation and journey time reliability metrics using automated fare data. Master's thesis, Massachusetts Institute of Technology.

Hamdouch, Y. and Lawphongpanich, S. (2008). Schedule-based transit assignment model with travel strategies and capacity constraints. *Transportation Research Part B: Methodological*, 42(7):663–684.

Koutsopoulos, H. N., Noursalehi, P., Zhu, Y., and Wilson, N. H. (2017). Automated data in transit: Recent developments and applications. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*, pages 604–609. IEEE.

Langlois, G. G., Koutsopoulos, H. N., and Zhao, J. (2016). Inferring patterns in the multi-week activity sequences of public transport users. *Transportation Research Part C: Emerging Technologies*, 64:1–16.

Nuzzolo, A., Crisalli, U., and Rosati, L. (2012). A schedule-based assignment model with explicit capacity constraints for congested transit networks. *Transportation Research Part C: Emerging Technologies*, 20(1):16–33.

Nuzzolo, A., Russo, F., and Crisalli, U. (2001). A doubly dynamic schedule-based assignment model for transit networks. *Transportation Science*, 35(3):268–285.

Ortega-Tong, M. A. (2013). Classification of london's public transport users using smart card data. Master's thesis, Massachusetts Institute of Technology.

Pelletier, M.-P., Trpanier, M., and Morency, C. (2011). Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568.

Poon, M., Wong, S., and Tong, C. (2004). A dynamic schedule-based model for congested transit networks. *Transportation Research Part B: Methodological*, 38(4):343–368.

Sumalee, A., Tan, Z., and Lam, W. H. (2009). Dynamic stochastic transit assignment with explicit seat allocation model. *Transportation Research Part B: Methodological*, 43(8):895–912.

Transport for London (2016). *WiFi trial to help give customers better journeys.*

Zhao, J., Rahbee, A., and Wilson, N. H. (2007). Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. *Computer-Aided Civil and Infrastructure Engineering*, 22(5):376–387.

Zhao, J., Zhang, F., Tu, L., Xu, C., Shen, D., Tian, C., Li, X.-Y., and Li, Z. (2017). Estimation of passenger route choice pattern using smart card data for complex metro systems. *IEEE Transactions on Intelligent Transportation Systems*, 18(4):790–801.

Zhu, Y. (2017). *Passenger-to-Itinerary Assignment Model Based on Automated Data*. PhD thesis, Northeastern University.

Zhu, Y., Koutsopoulos, H. N., and Wilson, N. H. (2017). Inferring left behind passengers in congested metro systems from automated data. *Transportation Research Procedia*, 23:362–379.

An equilibrium service choice in a dynamic traffic assignment with real-time information

Nam H. Hoang **Hai L. Vu**

Monash Institute of Transport Studies, Monash University, Australia

Email: {nam.hoang,hai.vu}@monash.edu

Dong Ngoduy

University of Canterbury, New Zealand

Email: dong.ngoduy@canterbury.ac.nz

October 9, 2018

1 Introduction

With the advancement of technologies, information becomes more accessible with improved quality and diversity (e.g. via sensors, phones and smart devices, or social networks, etc.). The literature review by Balakrishna et al. [1] showed important impacts of information services, e.g., the advanced traveller information systems (ATIS), to users and system performance, i.e.,

(a) Increased information's quality (e.g., updated frequency and perception variation) improves the system performance.

(b) The system performance is also improved as the market penetration increases, and then becomes stable after 50% penetration.

(c) Guided or equipped users gain most benefit in low market penetrations (i.e. below 30%), then start to lose some in the high penetration.

(d) Unequipped users mostly gain benefit at any positive penetration.

As several information providers like Google Map support free services to general users, travellers are not willing to pay for the premium services unless there is a considerable improvement of travel cost. According to the above fact, the pay-as-you-go services are more reasonable for users to economically utilise the real-time information. Even though the role and impact of information services have been studied extensively in the literature, most of them base their models on a given

market penetration. In this paper, we aim to investigate the elastic users of information services (or variation of penetration) where the traveller would prefer to use the service only if their gain of travel time is large in the comparison with the uninformed users.

Due to the elasticity of the informed users, we study the problem of equilibrium service choice in a mixed-user dynamic traffic assignment (ES-MUDTA) with real-time information. Particularly, the uninformed travellers follow the user equilibrium (UE) route choices before they enter the network, while the informed ones follow the system optimal (SO) choices with the capability of en-routing during their journey. There are several reasons that informed users would be more cooperative than the uninformed ones. Firstly, van Essen et al. [4] shows an increasing trend of bounded rational and non-selfish (social) route choices because the service providers could propose several options for users to, for example, save the fuel cost or eco-friendly reduce the pollution (in addition to the shortest paths). These alternative choices lead the system to operate closely to the SO solutions. Secondly, point (b) in the first paragraph indicates that the operators only need a fraction of users that commit to the guided instruction, to obtain a high or full achievable system performance. However, Gao [2] shows that this fraction could rise up to 60% in the UE solution of en-routing route choice. By using the system optimal choice, we expect the decrement of required committing users which certainly benefit the operators. Lastly, the future of connected and autonomous vehicles liberates drivers behind the wheels, therefore, avoids their cognitive expectation on route choices. It is reasonable that these smart cars will cooperatively work together to bring the best benefit to the system without sacrificing significantly the individual benefit.

In the following sections, we briefly describe the formulation of the ES-MUDTA problem, which is based on the link transmission model-LTM [5] for single-destination networks.

2 Problem formulation

The transportation network is represented by a directed graph $\mathcal{G} = (\mathbb{V}, \mathbb{A})$ where $\mathbb{V}$ is the set of vertices, and $\mathbb{A}$ is the set of directional arcs (or links). Let $\mathbb{A}_R$ denote the set of arcs from sources, $\mathbb{A}_S$ denote the set of arcs to sinks, and $\mathbb{A}_I$ denote the remaining arcs. Note that, the arcs in A_R and $\mathbb{A}_S$ are the virtual links that store traffic at sources and destinations to maintain the rule of flow conservation. The set of all possible paths, connecting source r and destination s in this network, is denoted by $\mathbb{P}^{(rs)}$. Let $\mathbb{T}$ denote a set of discrete times where T is the maximum time horizon, i.e., $\mathbb{T} = \{1, 2, \ldots, T\}$. Let Υ_a^- denote the set of inflow links to link a, and Υ_a^+ denote the set of outflow links from link a for any $a \in \mathbb{A}$. For any network topology in this paper, we have $\Upsilon_a^- = \emptyset \ \forall a \in \mathbb{A}_R$, and $\Upsilon_a^+ = \emptyset \ \forall a \in \mathbb{A}_S$.

The set of demand scenarios (or profiles) is denoted by X. Each scenario $x \in X$ has the probability ρ_x such that $\sum_{x \in X} \rho_x = 1$. Let $D_{rs,t|x}$ denote the amount of traffic demand at time

t from the source r to the destination s in the scenario x. For each link $a \in \mathbb{A}$, its characteristics are follows: L_a for length (m), K_a for vehicle density (veh/m), V_a for free-flow speed (m/s), W_a for backward-propagation speed (m/s), Q_a for flow capacity (veh/s). The set of user classes is denoted as $\mathbb{M} = \{i, n\}$, where the letter i represents the informed users, and the letter n represents the uninformed users. The flow variable $f^m_{ab,t|x}$ represents the amount of traffic belonging to the user-class m from link a to link b at time t in the scenario x. In the ensuing paper, we describe the formulation of LTM as linear constraints in the DTA problem, the information-based routing model and the overall model.

2.1 LTM-based constraints

In this part, we present the LTM in each scenario $x \in X$ for the network loading as a set of side constraints in the DTA problem. For further details of the linear formulation of the LTM-type constraints, we refer to the work [3]. The constraints of free-flow movement, backward shock-wave and flow capacity are shown below:

$$\sum_{b\in\Upsilon_a^+}\sum_{h\le t} f^m_{ab,t|x} \ge \sum_{b\in\Upsilon_a^-}\sum_{h\le t-L_a/V_a} f^m_{ba,t|x} \qquad \forall a\in\mathbb{A}, t\in\mathbb{T}, x\in X \qquad \text{(C.1)}$$

$$\sum_{m\in\mathbb{M}}\sum_{b\in\Upsilon_a^-}\sum_{h\le t} f^m_{ba,t|x} \le K_aL_a - \sum_{m\in\mathbb{M}}\sum_{b\in\Upsilon_a^+}\sum_{h\le t-L_a/W_a} f^m_{ab,t|x} \qquad \forall a\in\mathbb{A}, t\in\mathbb{T}, x\in X \qquad \text{(C.2)}$$

$$\sum_{m\in\mathbb{M}}\sum_{b\in\Upsilon_a^-} f^m_{ba,t|x} \le Q_a \qquad a\in\mathbb{A}, t\in\mathbb{T}, x\in X \qquad \text{(C.3)}$$

$$\sum_{m\in\mathbb{M}}\sum_{b\in\Upsilon_a^+} f^m_{ab,t|x} \le Q_a \qquad a\in\mathbb{A}, t\in\mathbb{T}, x\in X. \qquad \text{(C.4)}$$

For the FIFO constraints, we first denote $f^m_{ab,t,h|x}$ be the amount of traffic towards the destination s that enters link a at time t and entering link b at time h in the scenario x. Let $n^s_{a,t,h|x}$ be the amount of traffic towards the destination s that enters link a at time t and remains in this link at time h in the scenario x. The FIFO constraints are follows:

$$f^m_{ab,t,h|x} = \pi_{ab,t,h|x} n^m_{a,t,h-1|x} \qquad \forall a,b\in\mathbb{A}; t,h\in\mathbb{T}; x\in X \qquad \text{(C.5)}$$

$$(n^m_{a,t,h|x} - n^m_{a,t,h-1|x}) \sum_{k<t} n^m_{a,k,h|x} = 0 \qquad \forall a\in\mathbb{A}; t,h\in\mathbb{T}; x\in X. \qquad \text{(C.6)}$$

The above variables are defined below:

$$n^m_{a,t,h|x} = \begin{cases} 0 & \text{if } h < t + L_a/V_a \\ \sum_{b\in\Upsilon^-_a} f^m_{ba,t|x} & \text{if } h = t + L_a/V_a \\ \max(n^m_{a,t,h-1|x} - \sum_b f^m_{ab,t,h|x}, 0) & \text{if } h > t + L_a/V_a \end{cases} \tag{C.7}$$

$$\forall a \in \mathbb{A}, t \in \mathbb{T}, x \in X$$

$$f^m_{ab,h|x} = \sum_{t \le h - L_a/V_a} f^m_{ab,t,h|x} \qquad \forall a, b \in \mathbb{A}, h \in \mathbb{T}, x \in X. \tag{C.8}$$

2.2 Information-based routing

Real-time information Let $X_{t|x}$ be the set of possible scenarios realised at time t in scenario x. In this study, we assume the homogeneity of travellers (that they follow the same routing strategies) and the consistency of information (that it is provided equally to any travellers) in each user class. Therefore, the information at time t in scenario x, denoted as $y_{t|x} = \{X_{\tilde{t}|x}, \tilde{t}\}$ $(\tilde{t} \le t)$, represents

- The updated traffic states up to time $\tilde{t}$, i.e., $\{f_{ab,t|x'} : \forall a, b \in \mathbb{A}; s \in \mathbb{A}_S; t \le \tilde{t}; x' \in X_{\tilde{t}|x}\}$,
- The updated uncertainty of traffic demand up to time $\tilde{t}$, i.e., $X_{\tilde{t}|x}$.

We further assume that the quality of information is improved over time, i.e.,

$$X_{t|x} \subseteq X_{t-1|x} \tag{1}$$

for all $x \in X, t \in \mathbb{T}$. By writing $X_{y_{t|x}}$, we mean $X_{\tilde{t}|x}$, i.e., $X_{y_{t|x}} = X_{\tilde{t}|x}$. According to the definition of real-time information $y_{t|x}$, travellers receive the same information at time t_1 and t_2 in scenarios x_1 and x_2 respectively if $y_{t_1|x_1} = y_{t_2|x_2}$, which also means that $y_{t_1|x_1} = \{X_{\tilde{t}|x_1}, \tilde{t}\}$, $y_{t_2|x_2} = \{X_{\tilde{t}|x_2}, \tilde{t}\}$, $X_{\tilde{t}|x_1} = X_{\tilde{t}|x_2}$. The information-based flow split reads:

$$f^m_{ab,t|x'} = f^m_{ab,y_{t|x}} \tag{C.9}$$

for all $x' \in X_{y_{t|x}}$. It shows that if travellers receive the same information at time t_1 and t_2, i.e., $y_{t_1|x_1} = y_{t_2|x_2}$, then they follow the same traffic split.

Estimation of the travel cost $T^n_{p,y_{t|x}}$ of uninformed users Let $T^n_{p,t|x}$ denote the travel time on path p for uninformed traffic demand departing the source at time t in scenario x. The computation of $T^n_{p,t|x}$ is based on the departure flow at the source and the arrival flow at the destination along the path. Therefore, we compute $T^n_{p,y_{t|x}}$ from $T^n_{p,t|x}$ as follows:

$$T^n_{p,y_{t|x}} = \frac{1}{\sum_{x'\in X_{y_{t|x}}} \rho_{x'}} \sum_{x'\in X_{y_{t|x}}} \rho_{x'} T^n_{p,t|x'}. \tag{C.10}$$

Note that, $f^n_{p,t|x} = f^n_{p,t|x'} \quad \forall x' \in X_{y_{t|x}}$. Let $\tau^n_{a,y_{t|x}}$ be the estimation of the average travel time to destination s at the downstream of link a for a given information $y_{t|x}$. The conditions of UE path choices are follows:

$$T^n_{p,y_{t|x}} \geq \tau^n_{y_{t|x}} \qquad \forall p \in \mathbb{P}^{(rs)} \tag{C.11}$$

$$f_{p,y_{t|x}}(T^n_{p,y_{t|x}} - \tau^n_{y_{t|x}}) \leq 0 \qquad \forall p \in \mathbb{P}^{(rs)}. \tag{C.12}$$

Elastic informed users Given the fixed demand in each scenario, let $d^i_{rs,t|x}$ and $d^n_{rs,t|x}$ be the amount of informed and uninformed traffic respectively. Therefore, the conservation of traffic demand reads,

$$d^i_{rs,t|x} + d^n_{rs,t|x} = D_{rs,t|x} \tag{C.13}$$

for all $t \in \mathbb{T}, x \in X$, and the O-D pair (r, s). We hypothetically assume that a user is willing to pay for the information service if the difference of average travel time between informed users ($\tau^i_{y_{t|x}}$) and uninformed users ($\tau^n_{y_{t|x}}$) is at least β. This condition is presented below:

$$\min\{\tau^i_{y_{t|x}} + \beta, \tau^n_{y_{t|x}}\} \geq \tau^*_{y_{t|x}} \geq 0 \tag{C.14}$$

$$d^i_{rs,t|x}(\tau^i_{y_{t|x}} + \beta - \tau^*_{y_{t|x}}) \leq 0 \tag{C.15}$$

$$d^n_{rs,t|x}(\tau^n_{y_{t|x}} - \tau^*_{y_{t|x}}) \leq 0 \tag{C.16}$$

for all $t \in \mathbb{T}, x \in X$, given the O-D pair (r, s). In this paper, Eqs. (C.15) and (C.16) are also called the *conditions of equilibrium service choice*, and $\tau^*_{y_{t|x}}$ is the equilibrium service cost.

2.3 The overall model

We propose the following model for the mixed-user DTA problem as follows:

$$\begin{array}{ll} \text{Objective:} & \max F = \sum_{x\in\mathcal{X}} \sum_{t\in\mathbb{T}} \sum_{a\in\Upsilon^-_s} \rho_x(T+1-t)(f^i_{as,t|x} + f^n_{as,t|x}) \\ \text{s.t.} & \text{Constraints Eqs. (C.1) - (C.16).} \end{array}$$

References

[1] Ramachandran Balakrishna, Moshe Ben-Akiva, Jon Bottom, and Song Gao. Information impacts on traveler behavior and network performance: State of knowledge and future directions. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pages 193–224. Springer, 2013.

[2] Song Gao. Modeling strategic route choice and real-time information impacts in stochastic and time-dependent networks. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3): 1298–1311, 2012.

[3] D Ngoduy, N. H Hoang, H. L Vu, and D Watling. Optimal queue placement in dynamic system optimum solutions for single origin-destination traffic networks. *Transportation Research Part B: Methodological*, 92:148–169, 2016.

[4] Mariska van Essen, Tom Thomas, Eric van Berkum, and Caspar Chorus. From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport reviews*, 36(4):527–548, 2016.

[5] Isaak Yperman. The link transmission model for dynamic network loading (doctoral disseration). *KU Leuven, Belgium*, 2007.

The Vehicle Routing Problem with Load-Dependent Travel Times for Cargo Bike Routing

Pirmin Fontaine
School of Management
Technical University of Munich, Germany
Email: pirmin.fontaine@tum.de

1 Introduction

Many cities and urban areas already suffer under large amounts of traffic and congestion. Moreover, the growing e-commerce and the increasing population in cities further challenge the network. Therefore, under the name City Logistics, new concepts and business models are developed [1, 5]. Especially the concept of Two-Tier City Logistics Systems found a lot of attention recently [3]. One main idea in these systems is the use of environmental-friendly vehicles for the last mile delivery. One option for such an environmental friendly vehicle is the cargo bike. Recent studies showed that more and more companies are developing new cargo bikes and also that cities and delivery services are considering cargo bikes as an alternative for delivery [6]. But even if the transportation mode is changed from a diesel truck or van to a cargo bike, the problem still remains a Vehicle Routing Problem (VRP). While in the classical formulations, the travel time between two nodes is assumed to be constant, the travel times in the time-dependent VRP depend on the actual travel time if you want to reflect the different travel speeds and effects on emissions over the day [4].

To the best of our knowledge, all models assume a travel time that is independent of the vehicle load. So far, only load-dependent costs and emissions are considered in VRPs (e.g., [2]). Since the VRP is mostly used for scheduling trucks or delivery vans, the effect of the load on the speed is also negligible. Even for electric vehicles, the weight of the load takes only a small share in the total weight. However, when using cargo bikes for the final distribution of goods, the weight of the load is an important factor.

Therefore, we introduce the Vehicle Routing Problem with Load-Dependent Travel Times (VRPLDTT). As opposed to the classical VRP, we consider travel times that depend on the load of

the vehicle. We show how to calculate the possible speed depending on the weight and the slope of a street. Further, we formally define the VRPLDTT and introduce a new mixed-integer programming formulation. We define a new set of instances for the new problem setting, which is based on real cities. In an extensive numerical study, the importance of considering load-dependent travel times and influencing factors is shown.

2 Mathematical formulation

The VRPLDTT is defined on a graph $G = (N, A)$ with the set of customers $N_0 = \{1, \ldots, n\}$, the depot 0 ($N = N_0 \cup \{0\}$), and the set of arcs A. Each customer has a service time s_i, a demand q_i, and a time window $[a_i, b_i]$. d_{ij} defines the distance matrix and Q is the vehicle capacity.

The goal of our model is to minimize the total travel time of the bikers. The binary decision variable x_{ij} equals 1 if a vehicle drives on arc (i, j), and 0 otherwise. The continuous decision variable f_{ij} defines the load of the vehicle that is transported between node i and j. The arrival time at customer $i \in N_0$ is given by the continuous decision variable y_i and the number of vehicles leaving the depot by the positive integer variable is K.

Similar to the idea of speed levels as introduced by [2], we define a set of load levels $L = \{1, \ldots, l, \ldots\}$. Each load level corresponds to a load interval $[p^l, r^l]$ with $p^1 = 0$ and $r^{|L|} = Q$. Using this definition, we define the travel time in each interval $l \in L$ based on the average weight $(p^l + r^l)/2$ and the characteristics of the road segment (i.e., slope) as t^l_{ij}. The binary decision variable z^l_{ij} equals 1 if the travel time t^l_{ij} is used on arc (i, j) and 0 if not.

Using the introduced notation and decision variables, the problem is formulated as follows:

$$\min \sum_{(i,j)\in A} \sum_{l\in L} t^l_{ij} z^l_{ij} \tag{1}$$

subject to

$$\sum_{j\in N} x_{0j} = K \tag{2}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in N_0 \tag{3}$$

$$\sum_{i\in N} x_{ij} = 1 \qquad \forall j \in N_0 \tag{4}$$

$$\sum_{j\in N} f_{ji} - \sum_{i\in N} f_{ij} = q_i \qquad \forall i \in N_0 \tag{5}$$

$$q_j x_{ij} \leq f_{ij} \leq (Q - q_i) x_{ij} \qquad \forall (i, j) \in A \tag{6}$$

$$y_i - y_j + s_i + \sum_{l\in L} t^l_{ij} z^l_{ij} \leq M_{ij}(1 - x_{ij}) \qquad \forall i \in N, j \in N_0, i \neq j \tag{7}$$

$$a_i \leq y_i \leq b_i \qquad \forall i \in N_0 \tag{8}$$

$$\sum_{l \in L} z_{ij}^l = x_{ij} \qquad \forall (i,j) \in A \tag{9}$$

$$\sum_{l \in L} p^l z_{ij}^l \leq f_{ij} \leq \sum_{l \in L} r^l z_{ij}^l \qquad \forall (i,j) \in A \tag{10}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{11}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{12}$$

$$z_{ij}^l \in \{0,1\} \qquad \forall (i,j) \in A, l \in L \tag{13}$$

$$K \in \mathbb{N}_0 \tag{14}$$

The objective function is to minimize the total travel time. Constraint (2) ensures that K vehicles leave the depot. Constraints (3) and (4) ensure that each customer is visited exactly once. The flow balance constraints (5) guarantee that each customer demand is satisfied and the vehicle load is decreased after each customer visit. The vehicle capacity is ensured by constraints (6). Constraints (7) update the visiting times at each customer and together with constraints (8) ensure that the time windows are met. Constraints (9) state that a travel time level is only selected if a vehicle uses the arc. Since (9) further ensures that exactly one travel time level is selected for each traversed arc, constraints (10) guarantee that only the travel speed level that corresponds to the load weight is selected.

3 Load-dependent travel times

The power consumption when riding an (electric) bike depends on many factors. We assume that a cyclist drives on a straight line at constant speed in the considered segment. Then a cyclist has to face four main forces: the air resistance F_D, the rolling resistance F_R, the gravity when climbing up a hill F_G, and the friction of the mechanical parts F_F.

Besides physical constants and bike specific parameters, the power consumption depends on the following variable parameters: the speed of the bike v, the total mass m (of bike, biker, and cargo), and the slope of the street h. The total power is calculated as follows:

$$P = (F_D(v) + F_R(m,h) + F_G(m,h) + F_F)v \tag{15}$$

The detailed calculations will be shown during the presentation.

4 Numerical results and conclusions

We used the cities of Fukuoka (Japan), Madrid (Spain), Pittsburgh (USA), Seattle (USA), and Sydney (Australia) and placed a depot and 20 demand nodes in each city to generate a distance matrix using Google Maps. Further, different demand scenarios were generated which finally resulted in a total of 1,080 instances.

We analyze the effect of time windows and loading weight. We further show the benefits and compare them to classical models from the literature.

The results show that ignoring load-dependent travel times can lead to infeasible solutions in the presence of time windows. The consideration of load-dependent travel times can reduce the travel times by up to 23%. The benefits are particularly high if weights are low and the time windows are not too small, which reflects a typical last mile setting. The results show that a fitter cyclist can not only reduce the average travel time but also the battery consumption; even if the battery is the primary energy source. Moreover, the reduction of travel times comes with a reduction of energy consumption. Therefore, the model can provide parcel delivery services with more efficient options for routing. This is particularly important if you want to make the cargo bike a competitive alternative for a greener last mile delivery.

References

[1] T Bektaş, Teodor Gabriel Crainic, and Tom Van Woensel. From managing urban freight to smart city logistics networks. In K. Gakis and P. Pardalos, editors, *Networks Design and Optimization for Smart Cities*, volume 8 of *Series on Computers and Operations Research*, pages 143–188. World Scientific Publishing, 2017.

[2] Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

[3] Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Models for evaluating and planning city logistics systems. *Transportation science*, 43(4):432–454, 2009.

[4] Ola Jabali, T Woensel, and AG De Kok. Analysis of travel times and CO2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012.

[5] Martin Savelsbergh and Tom Van Woensel. City logistics: Challenges and opportunities. *Transportation Science*, 50(2):579–590, 2016.

[6] Susanne Wrighton and Karl Reiter. 5.3 Monitoring and evaluation report cyclelogistics ahead, 2017. URL `cyclelogistics.eu`.

Contributions of demand variability to unreliability in the public transport system

Emily Moylan
School of Civil Engineering
University of Sydney

Visiting Fellow at rCITI, UNSW Sydney

Email: emily.moylan@sydney.edu.au

1 Background

Across the range from microscopic simulation to regional strategy, traffic modelling relies on many assumptions about trip purpose, habitual behaviour, peak periods and scaling of impacts. Specifically, many analyses, such as strategic-level four-step models or cost-benefit assessment of a project will focus on the morning peak period in order to capture the system at maximum demand and simplify diversity of trip purpose and passenger demographic— modelling a complex and dynamic system becomes feasible by focusing on morning commuters.

But commutes are a minority of trips [1], not all trips are habitual [2], and the peak period may be too restrictive to accurately reflect the evolving labour ecosystem of shift work, flexible working arrangements, hot-desking, remote work, etc. Concerns about these and other assumptions in travel modelling have motivated the development of activity-based and dynamic approaches [3].

As new data becomes available, it is possible to validate the assumptions adopted in practice and quantify the extent to which these assumptions distort the modelling results. Notably, transit smart card data records habits and variations in travel longitudinally [4]. Furthermore, the importance of the assumptions should be particularly striking for public transit because systems are often optimised for a certain type of travel and traveller (habitual peak-period travel to and from the central business district (CBD)).

This paper explores demand variability using transit smart card data. The aim is to identify to what extent transit demand is influenced by predictable variation (time of day and day of week patterns) versus other phenomena. Assumptions about regularity are helpful for modelling, but they contribute to abstracting away variability and diversity in the system. This work quantifies the prevalence of repetitive travel, establishes a link between demand variation and system performance and explores the inter-dependencies between variation in public transport system components. In light of the findings,the discussion considers how some aspects of how those assumptions might contribute to misleading results at all modelling scales.

2 Repetitive travel on public transport

The following results are extracted from one month (April 2017) of transit smart card use in New South Wales (NSW), Australia. Each observation consists of a tap-on-tap-off pair with an anonymised card ID, date, time and location. The segments (defined by one tap-on and one tap-off) are chained into trips. Segments belong to the same trip if the same card taps on to the system within one hour of the previous tap-off. Repetition is defined when the same card ID travels between the same origin and destination more than once irrespective of time of day. Figure

1 shows the regularity of trips made in the first week of April 2017. There are 1,816,706 repeated trips. Over the same period, there are 2,110,525 unique cards being used.

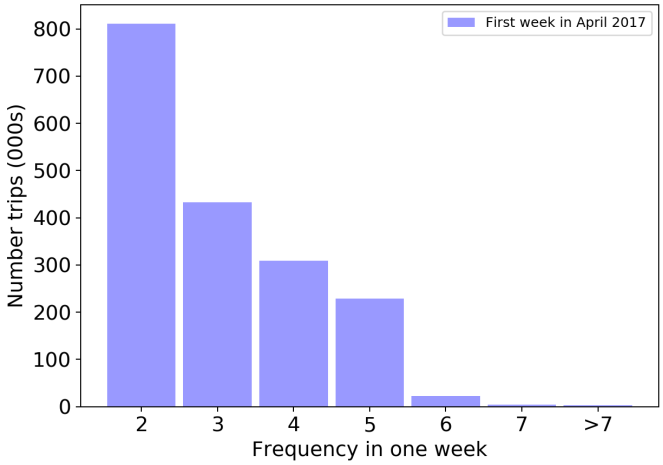


Figure 1: While repetition is extremely common, most trips do not match the picture of a commuter making the same journey five weekdays per week. Nearly half of all repeated trips only happen twice per week.

Without exogenous knowledge of the trip purpose, we define commute trips as those that occur more than three times in a week that start in the morning peak period (7-9am). For example, on 05 April 2017, there were 1,855,363 trips. 763,111 of these trips were repeated at least three times that week. Only 259,357 repeated trips occur in the AM peak. A model based on repeated (more than 3 times per week) trips occurring in the AM peak therefore captures only 13.9% of the daily travel on the NSW public transport system.

3 The link between passenger volume and system performance

Fluctuations in demand are expected. Figure 2 shows a pattern of predictable variations driven by days of the week and holidays. The remaining variability (day-to-day variability) is on the order of 100,000 trips per day. This variability is driven by diverse factors. For example, the comparison between number of trips and number of system users in Fig 2 shows that fluctuations in trip demand (important for an operational perspective) are influenced significantly by fluctuations in participation (number of users) as well as some variability in the intensity of participation, which explains why the number of trips and the number of users do not align perfectly.

Network operators can plan ahead for predictable variation though timetabling, route design and transfer coordination. Day-to-day variability in demand causes accompanying variability in the system performance. Figure 3 shows how the median travel time responds to changes in passenger volume for four journeys. Each observation represents a single hour in the month of April 2017 and the median travel times for all users making that trip is recorded. The fit is done on the weekday AM peak data (7-9am, a subset of the points shown) to control for timetable effects. All four journeys show how performance (inverse travel time) is anti-correlated with demand.

For the bus journey from Neutral Bay to Freshwater in Sydney's north, the longer travel times associated with high demand reflect road congestion (in-vehicle travel time) as well as delays at the transfer associated with bus bunching and crowding. The trip from Rozelle to Townhall has no transfers. Since bus-riders only tap-on when they board, it is not possible to measure bus bunching, schedule delay or crowding from this data.

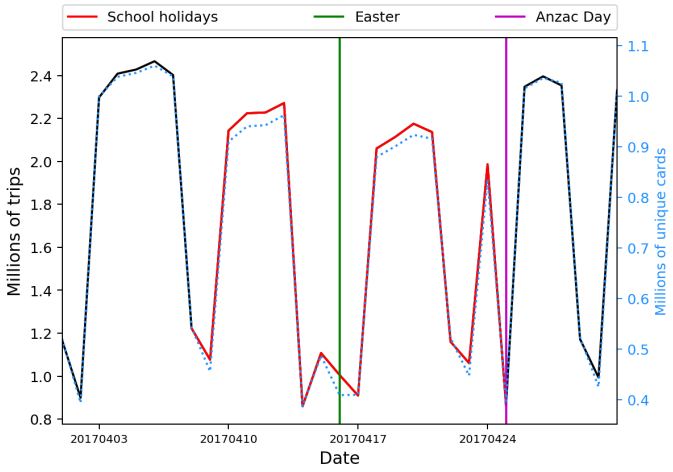
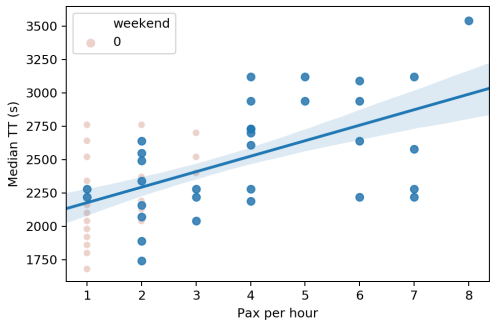


Figure 2: Number of trips per day over three months showing predictable (weekends, holidays, etc.) and unpredictable variations. The alternate vertical axis shows the variation in daily system users, which imperfectly mimics the pattern in the trip volumes.

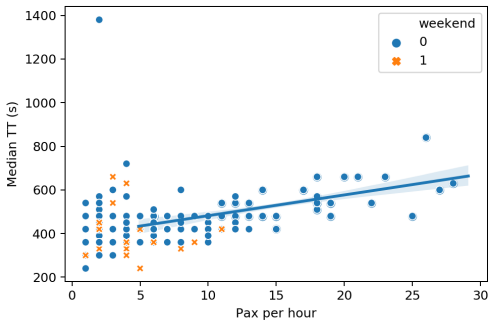
Train-riders tap when they enter the station, so their travel times include waiting time at the origin as well as transfer points. Because train transfers do not require tap-off-tap-on, it is unknown how many transfers occurred in a journey, although the ones shown here are likely to be direct services. In contrast with Lidcombe, Strathfield is a major transfer station with roughly double the demand to Central.
The trip from Lidcombe to Central highlights two attributes of the data. First, the single weekend observation with a high travel time is likely to be misleading. Because a trip is defined as the collection of segments where tap-on occurs within an hour of the previous tap-off, the trips that comprise this datum might include multiple waypoints that distort the actual travel time. Second, the segregation between weekend and weekday data highlights why the relationship is fit to weekday AM peak— travel times are slower when train services are less frequent or make more stops as they do on the weekend. The larger scatter in the weekend points also suggests a difference in travel behaviour between peak commuters and weekend travellers. These features both support the rationale of using simplifying modelling assumptions and illustrate their inaccuracies.

4 Proportionality and independence in variation between modes

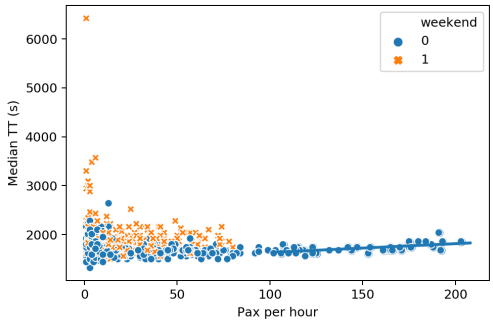
The phenomenon, shown for the bus journeys in Figure 3, that high passenger demands occur during periods of general congestion on the road suggests that variation across modes is correlated– the transport system as a whole varies, and its components vary proportionally. Transit smart card data contains information on the day-to-day variations of routes and modes as well as patterns in the types of users and their behaviours. Figure 4 shows how bus trips and train trips both follow time of day and day of the week patterns but exhibit independent variation. If demand in the transport system varied perfectly proportionally across the modes, the data in Figure 4b would lie on the fitted line. However, observations from the first part of the day (light colours) tend to have a lower than average fraction of train trips per hour whereas trains are slightly above average in the second half of the day. Moving through the day, the data trace a hysteresis loop on the plot. This dependency might be driven by behaviour associated with trip chaining (leisure and shopping activities located in rail-served activity centres are more likely to take place after work than before work). Day-to-day fluctuations in demand might also vary between train and bus because of differences in the service. For example, discretionary bus-riders might be more weather-sensitive than discretionary train-riders because bus stops tend to be more exposed than



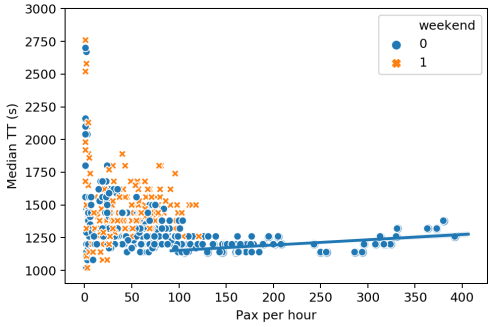
(a) Bus trip from Neutral Bay to Freshwater with a transfer in Manly.



(b) Bus trip from Rozelle to Town Hall with no transfers.



(c) Train trip from Lidcombe to Central.



(d) Train trip from Strathfield to Central.

Figure 3: Relationship between system performance and demand for different types of trips.

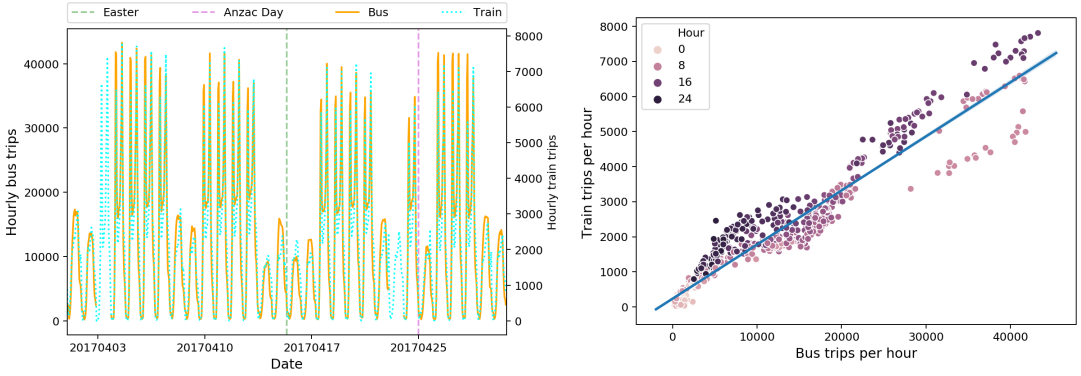
train stations. The complexities of the relationship between variations in the modes is removed by common simplifications and assumptions in travel modelling, but it leads to inaccuracies as fundamental as the ratio of train to bus demand.

5 Discussion

This work used transit smart card data to highlight nuances in variability in public transit demand. This evidence highlights a weakness in many of the modelling tools used in transport decision-making. Assessments of project benefits that rely on scaling of benefits from peak-period models or assume that all travel behaviour follows a archetypal commute pattern are likely to underestimate variability and undervalue a majority of trips. These weaknesses should be addressed to respond to growing awareness of the value of reliability and to consider the value of public transport more broadly. Due to the complexity of the transport system, simplifications are necessary to effectively assess hypotheticals and contingencies. However, the rising availability of data now makes it possible to quantify the likely impact of those assumptions and introduce more nuanced approaches where it counts.

5.1 Regular trips as a driver of consistent performance

One aspect of demand variability for future consideration is the habitual nature of travel. Commute trips are habit-driven and over-represented on public transit. 11.4% of all trips in the 2012-2013 New South Wales Household Travel Survey were on train or bus compared to 20.6% of work trips and 27.2% of education or childcare trips [1]. However, peak-period trips that repeat more than



(a) Number of trips per hour for train and bus showing the strong time of day and day of week patterns discussed above. However, the patterns are not strictly proportional.

(b) There are always more bus trips than train trips, but the relationship varies during the day with bus trips dominating in the morning and train trips in the evening and at night. The displayed fit is $Train = 0.15Bus + 236.20$.

Figure 4: Proportionality, independence and hysteresis in the variation in demand for trains and busses.

3 times per week account for less than half of all public transit trips (943,000/2,003,000 in the first week of April). Transit smart card data give a more functional view into trip regularity by identifying habitual trips regardless of purpose. Evidence of variation in regular trips will offer insight into existing literature on habitual travel [7, 6], contact networks [2] and learning behaviours such as after a home or work relocation [5, 8].

References

[1] Bureau of Transport Statistics, "Household Travel Survey Report: Sydney 2012/13", (November 2014). Available: `https://www.transport.nsw.gov.au/sites/default/files/media/documents/2017/HTS%20Report%20Sydney%202012-13.pdf`.

[2] Sun, L., Axhausen, K. W., Lee, D. H., Huang, X. (2013). Understanding metropolitan patterns of daily encounters. Proceedings of the National Academy of Sciences, 110(34), 13774-13779.

[3] Lam, William HK, and Yafeng Yin. (2001). An activity-based time-dependent traffic assignment model. Transportation Research Part B: Methodological, 35.6, 549-574.

[4] Pelletier, M. P., Trépanier, M., Morency, C. (2011). Smart card data use in public transit: A literature review. Transportation Research Part C: Emerging Technologies, 19(4), 557-568.

[5] Clark, B., Chatterjee, K., Melia, S. (2016). Changes to commute mode: The role of life events, spatial context and environmental attitude. Transportation Research Part A: Policy and Practice, 89, 89105. https://doi.org/10.1016/j.tra.2016.05.005

[6] Grling, T., Axhausen, K. W. (2003). Introduction: Habitual travel choice. Transportation, 30(1), 111. https://doi.org/10.1023/A:1021230223001

[7] Hanson, S., Huff, O. J. (1988). Systematic variability in repetitious travel. Transportation, 15(12), 111135. https://doi.org/10.1007/BF00167983

[8] Walker, I., Thomas, G. O., Verplanken, B. (2015). Old Habits Die Hard: Travel Habit Formation and Decay During an Office Relocation. Environment and Behavior, 47(10), 10891106. https://doi.org/10.1177/0013916514549619

Experimental Study of Congestion Pricing and the Role of Public Information

Mingyue Sheng

Energy Centre, Faculty of Business and Economics

The University of Auckland

Siwen Pan

Department of Economics, Faculty of Business and Economics

The University of Auckland

Mingyue Sheng

Energy Centre, Faculty of Business and Economics

The University of Auckland, Grafton Road, Auckland, New Zealand

Email: m.sheng@auckland.ac.nz

The aim of the research is to observe if congestion charges and public information given to participants can improve efficiency of the transport system, by conducting a laboratory experiment on a modified Market Entry Game (MEG), which represents the traffic coordination problem. Specifically, we will use a laboratory experiment to examine whether a congestion pricing mechanism, or toll, could reduce congestion of a traffic network with two routes, to its socially optimal level. In particular, we are interested in exploring four empirical issues. We intend to investigate: 1) whether a toll can reduce congestion, 2) whether there is a high variance in entry rates after implementing the toll, 3) whether the provision of public travel information before entry can reduce the variance level and improve efficiency, and 4) whether the group size of participants matters for entry decision.

The interest in using economic experiments to address whether a congestion tax is necessary is that it allows researchers to disentangle the specific impact that each dimension of the problem has in determining a given outcome. Laboratory settings allow for a controlled environment, within which to conduct much cleaner social engineering experiments. This is paradoxically so, as data generated in an experimental setting are not affected by having to determine any causality relationships between variables of interest, and are akin to eliminating any endogeneity problems as opposed to data extracted from real world situations. A number of experiments have investigated route-choice, including: [1],[2], [3], [4], [5],[6] and [7].

In this experiment based on a MEG, and following [8] and [9], a two-route congestion game is used to emulate a tragedy of the commons. Specifically, the subjects will face a two-route choice scenario in a controlled environment. In particular, a group of N commuters will choose, simultaneously and independently, between two routes: a slow reliable route and a faster, but

potentially congested route. Commuting time is fixed on the first route and is an increasing function of traffic on the second route. The average payoff an entrant benefits from is decrease in the number of entrants. This average payoff should be the same as the payoff from taking the slow reliable route. This equal-payoff equilibrium outcome is not socially optimal since entrants do not consider the effects of their own entry decisions on the other entrants. Therefore, the total payoffs with x entrants and N-x non-entrants is maximised when the marginal social benefit equals the marginal social cost. However, the variance of the entry rate also matters. This is so, because variance reduces welfare due to higher entry imposing external costs on more people, whereas fewer people get to enjoy any savings from lower entry. Therefore, we use a MEG to mimic the decision of choosing to either drive on a faster highway but at an additional cost, or endure possible congestion with no extra cost.

We will have a number of sessions. Each session will take approximately 90 minutes, and no more than 120 minutes. Some sessions will have 12 participants and some others will have 24 to check for robustness with regards to the numbers. Each session will involve 20 rounds of decision-making tasks. In each round, participants decide whether to enter the market. In every round in which participants did not enter, they will collect a fixed amount of experimental currency units (ECUs), i.e., converted, at a pre-specified ratio, into cash at the end of the experiment. Whereas, in every round in which participants did enter, they will collect an amount of ECUs determined by the total number of who else also entered in that round, to be also converted at the same pre-specified ratio, into cash at the end of the experiment. In this alternative scenario, the amount of ECUs for entry decreases with the number of the entrants, whereas the amount of ECU for exit is kept fixed. Furthermore, in the first 10 rounds, participants need not pay any fee to enter.

This is done to collect information about how individual decisions whether to enter are affected by the introduction of a toll, for example. In the remaining 10 rounds, participants need to pay a fee that corresponds to the level of what an optimal user fee should be, from a theoretical standpoint. Once again, this is done to see how participants respond to differing fee levels, particularly when contrasting their behaviour in the presence of a fee that is not necessarily the optional one with that behaviour when the fee is set at its optimal level, instead. This exercise will provide us with valuable insights regarding the entry rate, that is, whether it reaches the optimal level at any particular level of entry fees.

We also intend to vary the experimental sessions to allow for some treatments in which participants do not receive any of the collected fees, and other treatments in which instead such collected fees will be equally split among all participants, i.e., regardless of their idiosyncratic choices in the experimental session they participated in. The aim of such regime switch in the experimental setting is to see whether returning the collected fees to participants, a little like collected tolls on a road, could sponsor the provision of public goods providing participants with some form of rebate, would make them behave differently than when not returning such fees, thereby impacting differently on congestion.

Several more treatments will concentrate on the scenario with non-simultaneous entry decisions. We will let participants decide when to enter or exit, that is potentially while we show in real-time what the entry decisions of any other participants were, just like in the rush hour traffic reports. We will also let the participants decide the fees themselves in some additional sessions. Specifically, participants will vote on the level of the entry fee every 10 rounds and will split the fees

so collected. The purpose of this treatment is to check how close a voting process can be to an optimal or near-optimal fee levels. In theory, we expect to detect free-riding behaviour, of the kind associated with using roads as a public good. It would be interesting to see what the magnitude of such free-riding behaviour is in practice.

Furthermore, we will increase the group size from 12 to 24, to see if the findings are robust to larger groups (i.e. 24 participants) where no individual participant has as high of a direct influence on the congestion level, than when interacting with relatively smaller size groups (i.e. 12 participants).

Ultimately, at the end of each experiment, participants are asked to complete a questionnaire. The questionnaire involves a few demographic questions. Please see the attachment. Answers in the questionnaire are only meant to be used as control variables in our empirical analysis of the experimental data. It will only take 5 minutes to complete the questionnaire. Questionnaire will be printed on the paper and given to all participants at the end of the experiment.

Participants will be recruited from the University of Auckland through the Online Recruitment System for Economic Experiments (ORSEE), created by [10]. This is a web-based online recruitment system that is specifically designed for organising economic experiments. That includes sending email invitations via the system directly to the pool of participants who already registered, envisaging their participation in some future experiments conducted at the University of Auckland. Details of the system can also be found at www.orsee.org. Specifically, the ORSEE system allows to draw from a pool of 2,000+ students from all subjects, levels and faculties at the University of Auckland, and who voluntarily signed up to receive invitations to take part in various experiments held by researchers based at the University of Auckland. Participants in the experiments are invited through the system, by means of an e-invite. We provide a sample of such e-invite to this application, for further consideration. Following an invitation, those who are willing to take part in the lab experiments will be able to do so, by attending any of the scheduled experimental sessions they receive an invitation to. The experimental sessions are to be conducted at the Laboratory for Business Decision Making (DECIDE) at the University of Auckland Business School. We will use z-Tree by [11] to conduct computerised decision-making experiments. This is a widely used software package for developing and carrying out economic experiments. Details of the software can be found at www.ztree.uzh.ch. To keep the experiment unbiased, we will use a neutral environment rather than explicitly mentioning transportation, fuel tax or congestion price. All participants who are willing to take part in the lab experiments have to attend the experiment by person.

In conclusion, although it is quite natural to expect that congestion taxes will decrease the number of travellers, thereby increasing social welfare, and that the variance in entry rates will be high, quantifying those effects. Helping calibrate how varying levels of the tax and the precision of the information available to travellers remain of extreme importance. The significance of our results would be manifold. They would provide the necessary and scientifically based evidence to assist urban/transport planners and policymakers to gain a better understanding of how pricing schemes, and the provision of public travel information, will influence commuters' route-choice behaviour in New Zealand. The laboratory experiments will also offer a cost-effective way to identify market and policy drawbacks before any legislative changes.

References

[1] H.S. Mahmassani and G.-L. Chang, "Experiments with Departure Time Choice Dynamics of Urban Commuters", *Transportation Research Part B: Methodological*, 20(4), 297–320 (1986).

[2] Y. Iida, T. Akiyama and T. Uchida, "Experimental Analysis of Dynamic Route Choice Behavior', *Transportation Research Part B: Methodological* 26(1), 17–32 (1992).

[3] H.S. Mahmassani and Y.-H. Liu, "Dynamics of Commuting Decision Behaviour Under Advanced Traveler Information Systems", *Transportation Research Part C: Emerging Technologies* 7 (2–3), 91–107 (1999).

[4] D. Helbing, M. Schönhof, H.-U. Stark, H.-U. and J.A. Holyst, " How Individuals Learn to Take Turns: Emergence of Alternating Cooperation in a Congestion Game and the Prisoner's Dilemma", *Advances in Complex Systems* 8(2), 87–110 (2005).

[5] R. Selten, M. Schreckenberg, T. Chmura, T. Pitz, S. Kube, S.F. Hafstein, R. Chrobok, A. Pottmeier and J. Wahle, "Experimental Investigation of Day-to-Day Route-Choice Behaviour and Network Simulations of Autobahn Traffic in North Rhine-Westphalia", in *Human Behaviour and Traffic Networks*, M. Schreckenberg and R. Selten (eds), 1-21, Springer, Berlin, 2004.

[6] R. Selten, T. Chmura, T. Pitz, S. Kube and M. Schreckenberg, "Commuters Route Choice Behavior", *Games and Economic Behavior* 58(2), 394–406 (2007).

[7] L. Anderson, C. Holt and D. Reiley, "Congestion Pricing and Welfare: An Entry Experiment", in *Experimental Methods, Environmental Economics*, T.L. Cherry, S. Kroll and J.F. Shogren (eds), 280-292, Routledge, Oxon, 2008.

[8] R. Selten and W. Güth, "Equilibrium Point Selection in a Class of Market Entry Games", in *Games, Economic Dynamics and Time Series Analysis*, M. Diestler, E. Fürst and G. Schwadiauer (eds), 101-116, Physica-Verlag, Wien-Würzburg, 1982.

[9] R. Gary-Bobo, "On the Existence of Equilibrium Points in a Class of Asymmetric Market Entry Games", *Games and Economic Behavior* 2, 239-246 (1990).

[10] B. Greiner, "Subject Pool Recruitment Procedures: Organizing Experiments with ORSEE", *Journal of the Economic Science Association* 1(1), 114-125 (2015).

[11] U. Fischbacher, "z-Tree: Zurich Toolbox for Ready-Made Economic Experiments", *Experimental Economics* 10(2), 171-178 (2007).

Analytical BusPlus

Arthur Mahéo
Australian National University
Email: arthur.maheo@anu.edu.au

Michael Forbes
The University of Queensland
Email: m.forbes@uq.edu.au

Abstract

Analytical BusPlus is a framework for solving a multi-modal public tranportation network design problem using Benders Decomposition [Benders, 1962]. It incorporates state-of-the-art techniques for enhancing Benders and uses a novel technique to solve the Benders sub-problem. Our main contribution is a method to derive Pareto-optimal Benders cuts using an analytical procedure inside a branch-and-cut scheme.

Canberra is a planned city designed by American architect Walter Griffin in 1913. It features a large number of semi-autonomous towns separated by greenbelts. As a result, Canberra covers a wide geographic area, which makes public transportation particularly challenging. Bus routes are long and hence bus frequencies, and patronage, are low, especially during off-peak periods. To address these limitations, the BusPlus project designed, optimised, and simulated a Hub and Shuttle Public Transit System (HSPTS). The Hub and Shuttle model consists of a combination of a few high-frequency bus routes between key hubs and a large number of shuttles (or multi-hire taxis) that bring passengers from their origin to the closest hub and take them from their last bus stop to their destination.

A preliminary study for this project was conducted in Mahéo et al. [2017], by the first author. The study focused on designing a Hub-and-Shuttle Public Transit System, which is the problem of choosing a set of bus legs between pre-selected hubs in the city in order to minimise operation cost and maximise service convenience. The study presented a number of problem-specific pre-processing steps, then showed the advantage of using a Benders decomposition approach. In the study, the Benders decomposition was augmented using: a Pareto-optimal sub-problem, defined in Magnanti and Wong [1981], with a core point update policy from Papadakos [2008], and cut bundling using the problem structure [Birge and Louveaux, 1988]. We propose to extend the previous work by using an analytical framework to derive Pareto-optimal Benders cuts from the solution of the sub-problem and embed the cut generation process in a branch-and-cut framework.

The problem of linking a set of hubs using arcs is called the Hub-Arc Location Problem (HALP),

or hub-and-spoke network design problem. It was introduced by Campbell et al. [2005a,b] and is defined as the problem of locating a number of hub arcs in such a way that the total flow cost is minimised. As such, it is closely related to the well-known Hub Location Problem [O'Kelly, 1986]. The HALP is mostly used in transshipment contexts where economies of scale can be expected by grouping flows. Our formulation is very similar to Model HAL4 but we relax one important restriction: it is not necessary for paths to contain a hub-arc.

We modelled the HSPTS design problem as an Hub-Arc Location Problem. In the following, we use shuttles and taxis interchangeably, since the shuttles in our case study are multi-hire taxis, which are available in large numbers in Canberra. Bus routes can be opened for a fixed cost which represents the cost of operating high-frequency buses along the arc. The aim is to select among a number of hubs those that will form circular routes for buses. All other stops are reserved for shuttles. The objective is to minimise the cost of operating the system – i.e., the fixed cost of operating the bus lines and the variable cost for each taxi trip, together with maximising the convenience for the travellers. We use the trip duration as a proxy for traveller convenience in the model.

In the HSPTS, opening a bus leg is tantamount to opening an arc with a discounted flow. Thus, the HSPTS can be seen as a two-level decision problem: deciding which arcs to open first and then how to route the flow at minimum cost. As such, its structure appears ideally suited for Benders decomposition.

Our dataset represents a month's worth of trips in Canberra using the current public transit network. On average, weekdays have over 21,000 trips with around 60,000 passengers. The current bus network comprises about 2,800 bus stops, located on 94 bus lines. Each trip has an origin and a destination and a number of passengers. A time and distance matrix gives the on-road distance and average travel time between each pair of nodes, it is asymmetric and respects the triangle inequality. Finally, we have access to a pre-selected set of stops to base the bus network on.

Network problems are known to be highly degenerate. When modelling a network problem as a linear program, such as the sub-problem in BusPlus, this means its dual will admit many equivalent solutions. Because Benders relies on linear duality to generate cuts, we have to decide which dual solution to use. To enable an efficient choice Magnanti and Wong [1981] developed a procedure to generate "Pareto-optimal cuts." To derive such cuts, they propose solving two linear programs: the original sub-problem and a modified problem called a "Pareto sub-problem." We demonstrate how to derive dual costs from the primal solution of the Benders sub-problem in BusPlus and then prove these dual costs allow us to generate Pareto-optimal Benders cuts.

In general, for computing Benders cuts, we rely on a linear solver. In the case where we want to have Pareto-optimal cuts, this means solving two linear programs, which is computationally expensive. In BusPlus, the sub-problem is a shortest path. This problem can be solved to

optimality by dedicated algorithms faster than by using a general purpose linear solver. We propose an analytical framework to derive Benders cuts from the primal solution of our sub-problem.

Our idea to generate dual costs is to use their *natural interpretation*. This means that from the primal solution we should be able to derive the dual costs. At each Benders iteration, for each trip, we want to find the shortest path on the graph composed of:

- the union of the trip's origin and destination nodes with the potential hubs;
- the arcs selected at the current master iteration.

A summary of the interpretation of the dual costs associated with the nodes and arcs is as follows:

- For each node in the graph, the dual cost represents the potential savings achievable by going through the node.
- For each edge of the graph, which includes the closed arcs, the cost associated with an arc represents the potential savings incurred by opening the corresponding bus leg.

We provide results of three different setups: modeling and solving the HSPTS as a single MIP; solving the HSPTS using a tailored Benders decomposition; and, solving the HSPTS using our analytical Benders framework.

References

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962. doi: 10.1007/BF01386316.

John R. Birge and François V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392, 1988. doi: 10.1016/0377-2217(88)90159-2.

James F. Campbell, Andreas T. Ernst, and Mohan Krishnamoorthy. Hub Arc Location Problems: Part II—Formulations and Optimal Algorithms. *Management Science*, 51(10):1556–1571, oct 2005a. doi: 10.1287/mnsc.1050.0407.

James F. Campbell, Andreas T. Ernst, and Mohan Krishnamoorthy. Hub Arc Location Problems: Part I—Introduction and Results. *Management Science*, 51(10):1540–1555, oct 2005b. doi: 10.1287/mnsc.1050.0406.

Thomas L. Magnanti and Richard T. Wong. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29(3):464–484, 1981. doi: 10.1287/opre.29.3.464.

Arthur Mahéo, Philip Kilby, and Pascal Van Hentenryck. Benders Decomposition for the Design of a Hub and Shuttle Public Transit System. *Transportation Science*, 2017. URL `https://doi.org/10.1287/trsc.2017.0756`.

Morton E. O'Kelly. The Location of Interacting Hub Facilities. *Transportation Science*, 20(2): 92–106, may 1986. doi: 10.1287/trsc.20.2.92.

Nikolaos Papadakos. Practical enhancements to the Magnanti-Wong method. *Operations Research Letters*, 36(4):444–449, 2008. doi: 10.1016/j.orl.2008.01.005.

An activity-based approach for optimizing the High-Occupancy Toll lanes in congested road networks

Khoa D. Vo and William H. K. Lam

Department of Civil and Environmental Engineering

The Hong Kong Polytechnic University, Hong Kong

Emails: khoa.v.dang@gmail.com, william.lam@polyu.edu.hk

1 Introduction

Despite a large body of literature on the conventional network design models, very few work dealt with the network design problem for optimizing the High-Occupancy Toll (HOT) lanes, e.g. [1]. This study proposed a new activity-based approach for the captioned network design problem to investigate whether HOT lanes should be retrofitted into *existing* road links. The study hypothesizes that High-Occupancy Vehicles (HOVs) result from the joint travel of members within the same household. The motivation of intra-household joint travel is that household members often perform their daily activities and travel jointly with others [2, 3], often derived from household resource scarcities, such as vehicle allocation, and/or social, psychological and economic benefits [4].

The HOT optimization problem is formulated as a bi-level mathematical programing with equilibrium constraints (MPEC). The upper-level decision variables are related to whether HOT lanes are retrofitted into candidate road links and how much toll should be charged for Single-Occupancy Vehicles (SOVs). The lower-level decision variables are used to solved the household daily activity-based network equilibrium problem. Intuitively, the above bi-level problem is a mixed-integer optimization problem. Fortunately, it can be relaxed as a continuous problem by considering only continuous toll variables at the upper level. Then, a zero link toll for SOVs indicates no retrofit is necessary for the link, otherwise the retrofit is needed. It is also assumed that the budget for retrofitting HOT lanes into exisitng road links is negligible.

The new approach is based on the development of a new daily household activity-based network equilibrium model at the lower level, which takes into account the joint activity-travel scheduling behavior of household members. Household daily activity and travel choices are simultaneously represented by a joint activity-travel path (JATP) choice on a joint-time-space (JTS) supernetwork representation. The lower-level model is formulated as an equivalent variational inequality (VI) problem and solved by a heuristic solution method without the need to enumerate the feasible JATP choice set in advance.

2 Model formulation

2.1 Multi-lanes road network

Consider a road network $B = (S, A)$ where S is the set of nodes and A is the set of links. A node $s \in S$ can be a zone centroid or activity location. A link $a \in A$ represents lanes on a directed road link. Let A_1 and A_2, where $A_1 \cup A_2 = A$, denote of the sets of links representing General-Purpose (GP) and

HOT lanes, respectively. Let K be a finite set of discretized time intervals K. It is assumed that HOVs can travel on any lanes without toll while SOVs can only use HOT lanes with a toll. Let $\mathbf{y}$ be a toll vector, i.e. $\mathbf{y} = \{y_a(k) : a \in A_2, k \in K\}$ where $y_a(k)$ is the toll on link a at interval k. Then if $y_a(k) = 0$ indicates no HOT lane retrofit is necessary on link a at interval k, otherwise retrofit is needed.

Let J be the set of activities, such as at-home activities, work, waiting for pick-up and drop-off. Let H denote the set of household classes. For each household $h \in H$, let I^h denote the set of household members, and Z^h the set of all subsets of I^h including I^h itself but not the empty set. Then $z \in Z^h$ represents a group of persons of household h. Let Q^h denote the set of *feasible* JATPs for household $h \in H$, and P^{rs} the set of feasible paths from location $r \in S$ to location $s \in S$, which includes the sequence of links, i.e. either GP or HOT lanes, connecting r and s.

2.2 Joint activity-travel path

The concept of joint activity-travel path (JATP), suggested by [5], is adopted in this study to represent daily household activity and travel choices. A JATP choice is a set of interrelated decisions including (i) the trip chain and the car allocation for each member in the household, (ii) the activity location, start time, activity duration, and participating household members in each activity in the trip chain, and (iii) the path, departure time, travel time, and participating household members in each trip between two activity locations.

The daily (net) utility for household h using JATP q, denoted by U_q^h, is expressed as the difference between the total utility of daily activities participated in and the total disutility of daily travel of the household. That is,

$$U_q^h = \sum_k \sum_j \sum_s \sum_z U_{js}^{hz}(k)\xi_{js}^{hzq}(k) - \sum_k \sum_r \sum_s \sum_p \sum_z disU_{prs}^{hz}(k)\xi_{prs}^{hzq}(k), \quad \forall q, h, \tag{1}$$

where $U_{js}^{hz}(k)$ is the utility for group z of household h performing activity j at location s during interval k, $disU_{prs}^{hz}(k)$ is the disutility for group z of household h entering path p from r to s at interval k, $\xi_{js}^{hzq}(k)$ equals 1 if group z of household h using JATP q performing activity j at location s during interval k and 0 otherwise, and $\xi_{prs}^{hzq}(k)$ equals 1 if group z of household h using JATP q entering path p from r to s at interval k and 0 otherwise.

Note that in Eq. (1) group z of household h performs joint activity/travel if $|I^{hz}| \geq 1$ and solo activity/travel otherwise, where $|I^{hz}|$ is the number of persons in group z of household h. Then $|I^{hz}| \geq 1$ indicates the travel with a HOV and otherwise a SOV.

2.3 Household dail activity-based network equilibirum

Household members make activity-travel decisions to maximize household daily net utility. This leads to an equilibrium state at which no household can improve its daily utility by unilaterally changing its JATP choice to any other feasible one. The equilibrium condition is equivalent to the solution to the finite-dimensional variational inequality (VI) problem given by: finding a vector $\mathbf{f}^* \in \Omega$ such that

$$\sum_h \sum_q U_q^h(\mathbf{f}^*, \mathbf{y})\left[(f_q^h)^* - f_q^h\right] \geq 0, \quad \forall \mathbf{f} \in \Omega, \tag{2}$$

where f_q^h is the number of households with class h using JATP q, $\mathbf{f}$ is a vector of feasible JATP flows, i.e. $\mathbf{f} = \left\{(f_q^h) : \sum_q f_q^h = F^h, f_q^h \geq 0, q \in Q^h, h \in H\right\}$, and Ω the feasible region for feasible JATP flows at equilibrium, and F^h is the number of households with class h.

2.4 HOT lanes optimization problem

The HOT lanes optimization problem can be represented as a leader-follower, or Stackelberg game, where the HOT optimization is the leader, and household JATP equilibrium choices are the followers. Because building HOT lanes may reduce the number of SOVs but not the congestion level [6], the adopted objective function for the leader in this study is to minimize the total vehicle travel time instead of increasing car occupancies in the system. Then the interaction game can be represented as the following bi-level programming problem: finding a toll vector $\mathbf{y}^* \in R_+^n$, $n = |A_2| \times |K|$, and a flow vector $\mathbf{f}^* \in \Omega$ such that

$$(\mathbf{y}^*, \mathbf{f}^*) = \arg\min_{\mathbf{y},\mathbf{f}} \sum_k \sum_a t_a(k, \mathbf{f}) u_a(k) \tag{3}$$

subject to (2) and $u_a(k) \leq C_a, \forall a, k$, where $t_a(k, \mathbf{f})$ is the travel time (in intervals) on link a at interval k under $\mathbf{f}$, $u_a(k)$ is the vehicular flow entering link a at time interval k, and C_a is the capacity of link a (in vehicles/interval). The link inflow is derived by

$$u_a(k) = \sum_k \sum_h \sum_q \sum_z \sum_{r,s} \sum_a \sum_p \sum_{k' \geq k} f_q^h \varsigma_{prs}^{hz}(k) \xi_{apk}^{rs}(k') \quad \forall a, k,$$

where $\xi_{apk}^{rs}(k')$ equals 1 if vehicular flow entering path p from r to s at interval k' arrives link a at interval k and 0 otherwise.

3 Solution method

At the lower level, the VI problem (2) for the followers requires an explicit enumeration of feasible JATPs in advance. To avoid the need for enumerating feasible JATPs, a JTS supernetwork representation first is proposed such that a path in the supernetwork represents a feasible JATP. Thus the problem for searching the maximum utility JATP is transformed to a conventional path-finding problem in the supernetwork, which can be efficiently solved. Such a path-finding problem is integrated into the column generation procedure of the VI problem. A diagonalization method, based on the methods used in [7, 8], is then proposed to solved the VI problem. At the upper level, the continues HOT optimization problem is formulated as a mathematical programming with equilibrium constraint (MPEC), which can be efficiently solved by [9].

4 Discussion

Compared to trip-based approaches, such as [1], the two desirable features of our proposed activity-based approach at the lower level are (i) simultaneously considering of household activity and travel

choices when endogenously estimating car occupancies in the context of time-varying and network-wide congestion; and (ii) consistently modeling the number of persons in each car trip by the time of day to better reflect various levels of car occupancies and their impacts on travel cost and network performance. The above features facilitate the application of proposed model for a robust estimation of car occupancies and better evaluation of the ridesharing performance for the implementation of HOT lanes in practice.

Acknowledgment

The work described in this paper was jointly supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. 152057/15E and 15212217), and from the Research Committee of The Hong Kong Polytechnic University (Project No. 4-ZZFY).

References

[1] X. Di, R. Ma, H.X. Liu and X. Ban, "A link-node reformulation of ridesharing user equilibrium with network design", *Transportation Research Part B* 112, 230-255 (2018).

[2] J.P. Gliebe and F.S. Koppelman, "Modeling household activity–travel interactions as parallel constrained choices", *Transportation* 32, 449-471 (2005).

[3] P. Vovsha and E. Petersen, "Escorting children to school: Statistical analysis and applied modeling approach", *Transportation Research Record: Journal of the Transportation Research Board* 1921, 131-140 (2005).

[4] M.J. Roorda, J.A. Carrasco, and E.J. Miller, "An integrated model of vehicle transactions, activity scheduling and mode choice", *Transportation Research Part B* 43, 217-229 (2009).

[5] X. Fu and W.H.K. Lam, "Modelling joint activity-travel pattern scheduling problem in multi-modal transit networks", Transportation 45, 23-49 (2018).

[6] P. Murray, H. Mahmassani and K. Abdelghany, "Methodology for assessing high-occupancy toll-lane usage and network performance", *Transportation Research Part B* 1765, 8-15 (2001).

[7] H.K. Chen and C.F. Hsueh, "A model and an algorithm for the dynamic user-optimal route choice problem", *Transportation Research Part B* 32, 219-234 (1998).

[8] W.H.K. Lam and Y. Yin, "An activity-based time-dependent traffic assignment model", *Transportation Research Part B* 35, 549-574 (2001).

[9] S. Lawphongpanich and D.W. Hearn, "An MPEC approach to second-best toll pricing", *Mathematical Programming* 101.1, 33-55 (2004).

A Branch-and-Cut-and-Price Algorithm for the Capacitated Location-Routing Problem

Pedro Henrique P. V. Liguori
LAMSADE, Université Paris-Dauphine PSL Research University

A. Ridha Mahjoub
LAMSADE, Université Paris-Dauphine PSL Research University

Ruslan Sadykov
Inria Bordeaux — Sud-Ouest, Talence, France (Ruslan.Sadykov@inria.fr)

Eduardo Uchoa
LOGIS, University Federal Fluminense, Niteroi, Brazil

1 Introduction

In this work, we consider the standard *Capacitated Location-Routing Problem* (LRP), which is defined on a weighted undirected graph $G = (I \cup J, E \cup F)$. Vertices in I represent a set of possible depot locations, J denotes a set of customers. Edges $E = J \times J$ and $F = I \times J$ represent cheapest paths, with costs $c : E \cup F \to \mathbb{R}_+$, between pairs of vertices. Additionally, we associate capacities $W : I \to \mathbb{N}_+$ and opening costs $f : I \to \mathbb{R}_+$ with depots, and demands $d : J \to \mathbb{N}_+$ with each customer. There are identical vehicles with integer capacity Q. In this context, a *route* is an elementary cycle in G containing exactly one depot in I and a subset of the customers J. A LRP feasible solution is a set of routes such that: (i) each customer belongs to exactly one route; (ii) the sum of the demands of the customers in a route does not exceed Q; (iii) the sum of the demands of the customers in all routes associated to depot $i \in I$ does not exceed W_i. The goal is to find a feasible solution that minimizes the total route cost, the sum of the costs of the edges in each route, plus the opening costs of the depots used in the solution.

As observed by Contardo *et al.* [4], the LRP generalizes two important NP-hard problems: the *Capacitated Vehicle Routing Problem* (CVRP) and the *Capacitated Facility Location Problem* (CFLP). In fact, the integration of two levels of decisions, *i.e.*, location and routing, makes LRP an interesting model for several practical applications, from the design of telecommunications networks to the operation of very competitive supply chains. As already shown by Salhi and Rand [3], the

integration of location and routing decisions may lead to significant savings. We indicate Schneider and Drexl [6] as a recent survey on LRP.

2 Formulation with an exponential number of variables

Let Ω_i be the set of all routes associated with depot $i \in I$ that respect the capacity Q. For a set $K \subseteq I$, define $\Omega(K)$ as $\cup_{i \in K} \Omega_i$. Denote $\Omega(I)$ simply by Ω. For every $i \in I$, let y_i be a binary variable equal to 1 iff the depot i is opened. For every edge $(i,j) \in F$, let z_{ij} be a binary variable equal to 1 iff the customer j is served by depot i. Given $\omega \in \Omega$, let $a_e^\omega \in \mathbb{N}$ be a coefficient indicating how many times edge $e \in E \cup F$ is traversed by the route ω. If a route ω from depot i visits a single customer j, then $a_{ij}^\omega = 2$. For routes with two or more customers, $a_e^\omega \in \{0,1\}$, for $e \in \omega$. Finally, let λ_ω be a binary variable equal to 1 iff the route ω is used in the solution. Then the LRP can be formulated as

$$\min \quad \sum_{\omega \in \Omega} \left(\sum_{e \in E \cup F} c_e a_e^\omega \right) \lambda_\omega + \sum_{i \in I} f_i y_i \tag{1}$$

$$\text{subject to} \quad \sum_{i \in I} z_{ij} = 1 \qquad \forall\, j \in J, \tag{2}$$

$$\sum_{\omega \in \Omega_i} \sum_{e \in \delta(j)} a_e^\omega \lambda_\omega = 2 z_{ij}, \quad \forall\, i \in I,\ j \in J, \tag{3}$$

$$z_{ij} \leq y_i \qquad \forall\, i \in I,\ j \in J, \tag{4}$$

$$\sum_{j \in J} d_j z_{ij} \leq W_i y_i \quad \forall\, i \in I, \tag{5}$$

together with non-negativity and integrality constraints for all variables. Constraints (2) guarantee that every customer is served by exactly one depot. Constraints (3) are the degree constraints for customer nodes assuring that, if the customer j is served by depot i, then there must exist a route leaving depot i and passing through customer j. Constraints (4) imply that a customer can only be serviced by an opened depot and constraints (5) guarantee that the total demand supplied by the depot does not exceed its capacity.

For every $e \in E \cup F$ and every $i \in I$, let x_e^i be an additional integer variable counting how many times e is used by a route from depot i used in the solution. Variables x and λ are linked by the following identities:

$$x_e^i = \sum_{\omega \in \Omega_i} a_e^\omega \lambda_\omega, \quad \forall\, e \in E \cup F,\ \forall i \in I. \tag{6}$$

The integrality constraints on λ variables can be replaced by integrality constraints on x variables.

3 Valid inequalities

For a set $S \subseteq J$, define $d(S)$ as $\sum_{j \in S} d_j$. For a route $\omega \in \Omega$, $d(\omega)$ denotes the sum of the demands of customers visited by ω. For a set $K \subseteq I$, define $W(K)$ as $\sum_{i \in K} W_i$. The following family of inequalities can be obtained from the fact that capacity of the opened depots must be large enough to accommodate the whole demand. Define $\bar{y}_i = 1 - y_i$, for $i \in I$. Inequality $\sum_{i \in I} W_i y_i \geq d(J)$ is clearly valid and is equivalent to:

$$\sum_{i \in I} W_i \bar{y}_i \leq W(I) - d(J). \tag{7}$$

While (7) is redundant, the *Covering Inequalities* for the binary knapsack polytope defined by it are not redundant. These inequalities can be written as

$$\sum_{i \in C} (1 - y_i) \leq |C| - 1, \quad \forall\, C \subseteq I : W(C) > W(I) - d(J). \tag{8}$$

Additionally, we introduce a new family of valid inequalities. Let $\theta_i^q \in \mathbb{N}$ be a variable indicating how many routes with a total load of exactly q units leave depot i. Denoting $\Omega_i^q = \{\omega \in \Omega_i : d(\omega) = q\}$, then variable θ_i^q can be expressed as $\theta_i^q = \sum_{\omega \in \Omega_i^q} \lambda_\omega$. The following inequalities are valid for the LRP:

$$\sum_{q=1}^{Q} q \theta_i^q \leq W_i y_i, \quad \forall\, i \in I. \tag{9}$$

While those inequalities are redundant, they define integer knapsack polyhedra and can be used as source of violated cuts, that we call *Route Load Knapsack*. Those cuts were separated using the procedure proposed by Chopra *et al.* [2].

4 Algorithm and results

We have extended the Branch-and-Cut-and-Price Algorithm of Sadykov *et al.* [5] to solve formulation (1)–(5) reinforced by (8) and by the Route Load Knapsack derived from (9). The algorithm employs a number of techniques proved to be effective for solving classic Vehicle Routing Problems: *ng*-route relaxation, automatic dual price smoothing stabilization and enumeration of elementary routes. The branching here is performed on variables y, z and x. Multi-phase strong branching is used to select the candidate for branching. Additionally we separate Rounded Capacity Cuts and limited-memory set packing Rank-1 cuts. The bucket graph based labeling algorithm is used to solve the pricing problem and generate feasible routes. The new cutting planes we propose are *non-robust*, i.e. the labeling algorithm has been appropriately modified to take them into account.

Preliminary results showed that our algorithm could solve to optimality, for the first time, 12 open instances of the most difficult classes $\mathcal{F}_2$ and $\mathcal{F}_4$. These instances, containing up to 200 customers and 10 depot locations, could not be solved by the state-of-the-art approach by Contardo

et al. [4]. The only remaining open instance for class $\mathcal{F}_2$ is now 200-10-3b. In the table below we present the running times of the algorithm. The underlined optimum solution values improve the best known solutions from Schneider and Löffler [7].

Instance	Init. UB	Optimum	Time	Instance	Init. UB	Optimum	Time
100x5-1b	213570	213568	10m05s	200x10-1b	375180	375177	1h55m
100x10-1a	287670	287661	1h32m	200x10-2a	448080	<u>448005</u>	4h45m
100x10-1b	231000	230989	1h38m	200x10-2b	373700	373696	5h53m
100x10-3a	250890	250882	1h17m	P113112	1239.00	1238.24	2h29m
100x10-3b	203120	203114	11h01m	P131112	1893.00	1892.17	36m52s
200x10-1a	474860	<u>474702</u>	20m42s	P131212	1961.00	1960.02	34m59s

It should be noted that existing exact approaches for the LRP are based on enumeration of subsets of open depots and thus impractical for instances with more than 10 depot locations. Our algorithm is free of this drawback. It has already solved to optimality some instances of class $\mathcal{F}_4$ with 20 depot locations. No exact approach has been devised in the literature for such instances.

References

[1] J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. Wolfler-Calvo, "A Branch-and-Cut Algorithm for the Capacitated Location Routing Problem", in *Computers & Operations Research* 38, 931-941 (2011).

[2] S. Chopra and S. Shim and D. Steffy, "A few strong knapsack facets", Modeling and Optimization: Theory and Applications (MOPTA), at Lehigh University, USA (2014).

[3] S. Salhi, G.K. Rand, "The effect of ignoring routes when location depots", in *European Journal of Operations Research* 39(2), 150-156 (1989).

[4] C. Contardo, J.F. Cordeau, B. Gendron, "An exact algorithm based on cut-and-column generation for the Capacitated Location-Routing Problem", in *INFORMS Journal on Computing* 26(1), 88-102 (2014).

[5] R. Sadykov, E. Uchoa, A. Pessoa, "A Bucket Graph Based Labeling Algorithm with Application to Vehicle Routing", in *Cadernos do LOGIS* 7 (2017).

[6] M. Schneider, M. Drexl, "A Survey of the standard Location Routing Problem", in *Annals of Operations Research* 259(1), 389-414 (2017).

[7] M. Schneider, M. Löffler, "Large Composite Neighborhoods for the Capacitated Location-Routing Problem", in *Transportation Science* Published Online (2017).

TRISTAN X

Extended Abstract

Shared Autonomous Mobility Fleets and Multimodal Transit Networks: Design Methodology and Trade-Offs

Hani S. Mahmassani (Corresponding Author)
William A. Patterson Chair in Transportation
Director, Northwestern University Transportation Center
600 Foster St.
Evanston IL 60208
masmah@northwestern.edu

Helen Pinto
PhD Candidate
Northwestern University
HelenPinto2020@u.northwestern.edu

Michael Hyland
Assistant Professor
University of California, Irvine
hylandm@uci.edu

1 Introduction

Fully-autonomous vehicles (AVs) and shared autonomous mobility services (SAMSs) are expected to offer considerable cost advantages over existing driver-operated non-fixed-route mobility services (e.g. taxi, paratransit, demand-responsive transit), thus providing transit agencies the opportunity to redesign their entire service regions and service networks. This study is predicated on the premise that significant opportunity may exist for providing (or subsidizing/contracting third party) SAMSs in lieu of low-frequency transit service in low-density regions of a metropolitan area, redirecting resources towards high-frequency, high-quality rapid transit services, resulting in a win-win situation for transit agencies and transit users.

To test this premise, this study introduces the joint transit network (re)design and SAMS fleet size determination problem; presents a bi-level mathematical programming formulation; and outlines a solution approach for the bi-level mathematical program. This study models the upper-level problem via adapting a transit network frequency setting problem (TNFSP) formulation. The TNFSP formulation is adapted in this study to (1) allow possible removal of transit service patterns, and (2) incorporate the utility of SAMS users into the objective function. The lower-level problem is an integrated mode choice-traveler assignment problem that takes the SAMS fleet size and transit pattern frequencies as input and returns mode- and pattern-specific demand. The formulation for the lower-level problem is a gap-based fixed-point equilibrium formulation. Both the upper-level and the lower-level problems are analytically intractable; hence, this study develops and demonstrates an effective heuristic solution approach for large-scale network applications.

Features of the modeling framework presented in this paper include (i) capturing congestion (boarding rejections and seat/standing space availability) and transfers in the transit network; (ii) incorporating three transit modes: transit-only, SAMS-only, and transit-SAMS; (iii) extending the concept of route patterns first used in the TNFSP by (Verbas and Mahmassani, 2013) to include the frequency setting of limited-stop lines; (iv) considering spatial and temporal heterogeneity of demand in the lower level, while upper-level takes the demand supplied by the lower-level; (v) quantifying the response of modal shares due to the service changes.

Route patterns are subsets of ordered stops for a certain route and dispatch time. The application in TNFSP of this concept is extended by including all existing patterns of a route throughout the year as a potential pattern for each time interval, so that limited-stop lines are considered before the decision to fully eliminate a route. This extension provides a more complete analysis, with a larger feasible set of patterns. Direct benefits of limited-stop lines are higher likelihood of meeting demands and expected operator savings from shorter cycle times, as well as improvements in the level of service from the user's perspective because of reduced travel times (Ibarra-Rojas et al., 2015).

Few studies have included limited-stop lines in the TNFSP (Afanasiev and Liberman, 1983; Chiraphadhanakul and Barnhart, 2013; Freyss et al., 2013; Leiva et al., 2010; Ulusoy et al., 2011). Additionally, this work captures urban modal split response to the frequency setting of route patterns (including of limited-stop lines) using simulation, as part of the integrated mode choice and transit assignment-simulation modeling framework to support joint transit-SAMS service planning.

2 Modeling Framework

The joint transit network redesign and SAMS fleet size problem is modeled as a bi-level mathematical program. The generic formulation of a bi-level mathematical program is presented in Equations (1-2).

Upper Problem: $$\min_{x} F[x, y]\,;\; G[x, y] \leq 0 \quad (1)$$

Lower Problem: $$\min_{y} f[x, y]\,;\; g[x, y] \leq 0 \quad (2)$$

$F[\cdot]$: objective function of the upper-level decision makers
x: decision vector for the upper-level decision makers
$G[\cdot]$: constraint set of the upper-level decision vector
$f[\cdot]$: objective function of the lower-level decision makers
y: decision vector for the lower-level decision makers
$g[\cdot]$: Constraint set of the lower-level decision vector

$y = y(x)$ is typically referred to as the reaction or response function. The key to solving the bi-level programming model is to obtain the response function from the lower-level problem. Then the variable y in the upper-level problem can be replaced with the relationship between y and x (Sun et al., 2008).

The bi-level mathematical program can be seen as a Stackelberg game in which the leader is the transit-SAMS designer, and the followers are the travelers. In this study, the transit-SAMS designer sets the transit pattern frequencies (f_p), and the SAMS fleet size (S) with knowledge of how travelers will respond to a given design. Given the transit-SAMS design, travelers choose their modes and routes. The objective of the transit-SAMS designer is to maximize the utility of all travelers, whereas the objective of an individual traveler is to maximize their own utility.

The demand for each pattern and the SAMS mode is fixed in the upper-level problem and determined in the lower-level problem. The upper-level model determines the transit pattern frequencies, and the SAMS fleet size. This information is fed into the lower level problem which is an integrated mode choice-traveler assignment problem. The lower level problem returns the mode- and pattern-dependent demand/flow information to the upper level problem.

3 Mathematical Formulation

3.1 Upper-Level Formulation

Let p denote a pattern in the set of transit route patterns (P), wherein P_b denotes the subset of bus transit patterns, and P_r the subset of rail transit patterns. Let l_p be the length and d_p the dispatch time interval of pattern $p \in P$. The dispatch time interval (d_p) is a member of the set of time intervals T, indexed by $t \in T$. The lower-level problem determines the demand for each pattern e_p and the demand for the SAMS mode during time interval $t \in T$, e_{AV}^t.

The transit-SAMS designer aims to maximize the utility of all travelers, subject to a budget Γ. The designer can set the pattern frequencies f_p; determine whether to remove a pattern ($y_p = 1$) or keep a pattern ($x_p = 1$); and set the SAMS fleet size S. The upper-level problem formulation is presented in Equations (3-11).

$$\min_{f,x,y,S} \sum_{p \in P} \frac{x_p e_p}{2 f_p} + \sum_{t \in T} \sum_{\{p \in P_b | d_p = t\}} \left(y_p e_p + e_{AV}^t\right) \times w_o \left(1 + \alpha \left(\frac{y_p e_p + e_{AV}^t}{r_s S}\right)^{\beta}\right) \quad (3)$$

$$\sum_{\{p \in P_b | d_p = t\}} y_p e_p + e_{AV}^t \leq r_s S \times (1 + \gamma) \qquad \forall t \in T \quad (4)$$

$$c_2 \sum_p f_p l_p - c_2 \sum_p y_p f_{min} l_p + c_3 S \leq \Gamma \quad (5)$$

$$x_p f_{min} < f_p \qquad \forall p \in P \quad (6)$$

$$x_p + y_p = 1 \qquad \forall p \in P \quad (7)$$

$$f_p \geq f_{min} \qquad \forall p \in P \quad (8)$$

$$x_p = 1 \qquad \forall p \in P_r \quad (9)$$

$$S \geq 0 \quad (10)$$

$$x_p, y_p \in \{0,1\} \qquad \forall p \in P \quad (11)$$

where, w_o is the minimum average traveler wait time; r_s is the service rate of an AV; c_2 is a multiplier for the transit operational costs and c_3 is the cost per AV; and f_{min} is the minimum transit pattern frequency.

The objective function in Equation (3) aims to minimize the disutility of travelers. The first term represents the cumulative wait time of transit travelers that are assigned to transit patterns that remain ($x_p = 1$). Assuming travelers arrive randomly, the average traveler wait time on pattern $p \in P$ is $\frac{0.5}{f_p}$. Multiplying the average traveler wait time, by the demand for pattern $p \in P$ (e_p) gives the cumulative wait time of travelers using pattern $p \in P$.

The second term in the objective function represents the cumulative wait time of the travelers originally assigned to an SAMS (e_{AV}^t) and the travelers that were assigned to pattern that was removed ($y_p e_p$). The term $\sum_{\{p \in P_b | d_p = t\}} \left(y_p e_p + e_{AV}^t\right)$ represents the cumulative demand for SAMSs at time interval $t \in T$; whereas, $\sum_{\{p \in P_b | d_p = t\}} w_o \left(1 + \alpha \left(\frac{y_p e_p + e_{AV}^t}{r_s S}\right)^{\beta}\right)$ represents the average wait time of SAMS travelers during time interval $t \in T$. The parameter w_o represents the average SAMS traveler wait time if the SAMS fleet size (S) is large and the SAMS demand ($y_p e_p + e_{AV}^t$) is small. Holding fleet size constant, as the SAMS demand increases the average SAMS traveler wait time should steadily increase until the demand rate approaches the service rate, at which time the average SAMS traveler wait time should grow exponentially.

Equation (4) requires that the SAMS demand is not much greater than the AV fleet service capacity. Equation (5) is a budgetary constraint. The first term represents the operating cost of providing service on a pattern $p \in P$ of length l_p at a frequency f_p. The second term corrects for the patterns that are removed ($y_p = 1$). The reason this term is needed is because the frequency of a pattern cannot be set to zero due to Equation (8). If Equation (8) is removed the first term in the objective function would need to go to infinity to represent the removal of transit pattern. The third term in Equation (5) represents the cost of an SAMS fleet size S.

Equation (6) requires the pattern frequency f_p to be greater than the minimum frequency f_{min} if the pattern remains (x_p=1). Equation (7) requires a pattern to be removed (y_p=1) or not removed ($x_p = 1$). Equation (8) ensures that the pattern frequency of all patterns (f_p) is greater than or equal to the minimum pattern frequency (f_{min}). Equation (9) requires the rail transit patterns to remain. Equation (10) requires the fleet size to be positive and Equation (11) requires y_p and x_p to be binary.

3.2 Lower-Level Formulation

The lower-level formulation is an integrated mode choice-traveler assignment problem based on a formulation introduced in (Verbas et al., 2016).

4 Solution Approach

Both the upper-level and lower-level formulations of the bi-level transit network redesign and SAMS fleet size modal are analytically intractable; hence, we present a brief overview of a heuristic solution approach (see Figure 1). The algorithm begins by solving the traveler assignment problem given initial origin-destination-mode-departure time (ODMT) demand, an initial transit network, an initial set of transit pattern frequencies, and an initial SAMS fleet size. The SAMS simulator obtains experiences for individual travelers via running a simulation and dynamically operating an SAMS fleet, using advanced assignment, routing, and scheduling algorithms. The transit assignment-simulation model, solves a congested multi-modal time-dependent assignment problem via iteratively (1) determining least-cost transit hyperpaths on a time-dependent network; (2) assigning transit travelers to a transit hyperpath; and (3) simulating the performance of transit travelers and vehicles in a congested urban transit network. The transit-assignment simulation model returns the performance of the transit network and the experience of individual travelers.

The performances of the SAMS and the transit network at the ODMT-level are fed into a mode choice model. The mode choice model assigns or reassigns individual travelers to transit-only, SAMS-only, or SAMS-transit, based on the ODT performances of each mode. The mode choice model then feeds this demand into the traveler assignment module. This process repeats until the modal shares, and the mode choice probabilities converge. This is a challenging problem as the mode choice probabilities depend on the transit and SAMS system performance; yet, the transit and SAMS system performance depends on the modal shares. Hence, many iterations of the mode choice model are required to reach equilibrium.

The integrated mode choice-traveler assignment problem returns transit pattern-level demand and time-dependent SAMS demand to the transit-SAMS design module. With this information, the transit-SAMS design module solves the mathematical program displayed in Equations (3-11). This is a non-convex, non-linear, integer programming problem. Hence, a heuristic solution approach that efficiently explores the solution space is required to obtain good, feasible solutions (finding optimal solutions is highly unlikely). The transit-SAMS design module returns transit pattern frequencies, as well as the transit patterns that have been removed, along with the SAMS fleet size. This information is fed back into the integrated mode choice-traveler assignment module. This process repeats until the transit-SAMS design solution converges in terms of either the objective function or the decision variables.

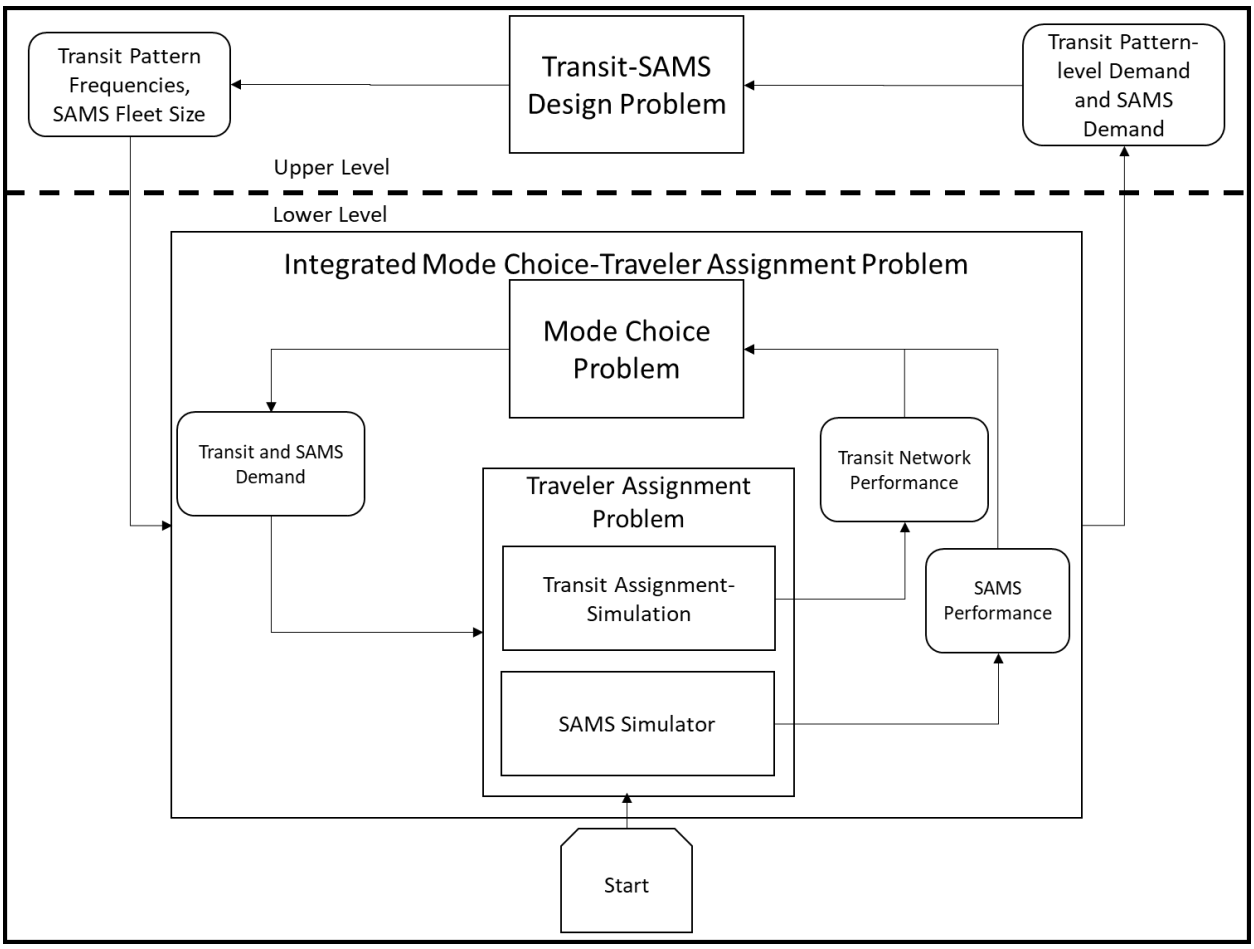


Figure 1: Algorithm to solve the bi-level transit network redesign and SAMS fleet size problem

5 Computational Results

The model is applied to the Chicago, Illinois area, featuring a large-scale multimodal transit system, as well as actual demand patterns calibrated for the existing network. The computational analysis will (i) test the ability of the solution algorithm to improve the utility of travelers; (ii) analyze the impacts of SAMSs on transit network design and transit patterns frequencies; (iii) test hypothesis that SAMSs will replace low-frequency, low-demand transit patterns; (iv) perform sensitivity analysis on relevant model parameters including AV cost and transit fare, as well as transit-SAMS joint fare; and (v) examine design trade-offs between conventional transit service and SAMs service from the standpoint of overall performance and user experience.

References

Afanasiev, L.L., Liberman, S.Y., 1983. Principles for organizing express bus services. Transp. Res. Part A Gen. 17, 343–346. doi:10.1016/0191-2607(83)90002-X

Chiraphadhanakul, V., Barnhart, C., 2013. Incremental bus service design: Combining limited-stop and local bus services. Public Transp. 5, 53–78. doi:10.1007/s12469-013-0067-7

Fan, W., Machemehl, R., 2011. Bi-Level Optimization Model for Public Transportation Network Redesign Problem. Transp. Res. Rec. J. Transp. Res. Board 2263, 151–162. doi:10.3141/2263-17

Freyss, M., Giesen, R., Muñoz, J.C., 2013. Continuous approximation for skip-stop operation in rail transit. Transp. Res. Part C Emerg. Technol. 36, 419–433. doi:10.1016/j.trc.2013.07.004

Leiva, C., Muñoz, J.C., Giesen, R., Larrain, H., 2010. Design of limited-stop services for an urban bus corridor with capacity constraints. Transp. Res. Part B Methodol. 44, 1186–1201. doi:10.1016/j.trb.2010.01.003

Sun, H., Gao, Z., Wu, J., 2008. A bi-level programming model and solution algorithm for the location of logistics distribution centers. Appl. Math. Model. 32, 610–616. doi:10.1016/J.APM.2007.02.007

Ulusoy, Y.Y., Chien, S.I.-J., Wei, C.-H., 2011. Optimal All-Stop, Short-Turn, and Express Transit Services Under Heterogeneous Demand. Transp. Res. Rec. J. Transp. Res. Board 2197, 8–18. doi:10.3141/2197-02

Verbas, I., Mahmassani, H., 2013. Optimal Allocation of Service Frequencies over Transit Network Routes and Time Periods. Transp. Res. Rec. J. Transp. Res. Board 2334, 50–59. doi:10.3141/2334-06

Verbas, I.O., Mahmassani, H.S., Hyland, M.F., Halat, H., 2016. Integrated Mode Choice and Dynamic Traveler Assignment in Multimodal Transit Networks: Mathematical Formulation, Solution Procedure, and Large-Scale Application. Transp. Res. Rec. J. Transp. Res. Board 2564, 78–88. doi:10.3141/2564-09

Applying Meta-heuristic Algorithm with parallel computation framework to simulation-based Dynamic Traffic Assignment

Mostafa Ameli [a,b]

14-20 Boulevard Newton, 77420 Champs-sur-Marne, France

Email: mostafa.ameli@ifsttar.fr

Jean-Patrick Lebacque [a]

Ludovic Leclercq [b]

[a] University of Paris-Est, IFSTTAR, GRETTIA

[b] Univ. Lyon, IFSTTAR, ENTPE, LICIT

1 Introduction

The Dynamic Traffic Assignment (DTA) refers to the procedure of assigning trips to paths in a given transportation system considering the Origin Destination pair (OD) flow demand and the network dynamic traffic states. The main output of DTA is path flow distribution over all feasible paths for all OD pairs. Travelers in the traffic network usually attempt to minimize their own travel time (cost). The solution of the assignment problem which is based on Wardrop's first principle is called User Equilibrium (UE).

The goal of this study is computing UE solutions in a simulation-based DTA process. There are many researches that have shown this problem can be represented as a fixed-point problem [1]. There are many solution algorithm in the literature that have been proved to be efficient to solve DTA. Nevertheless, in a large-scale and trip-based setting it is not possible to guarantee that fixed point algorithms converge towards the optimal solution because of the step size and because there is no exact method for determining the step size in the literature [2]. There are some criteria such as the total gap [3] to see how far the solution is from the optimal solution. It often happens that the total gap stops improving after some iterations because the step size is small and prevents the solution from being improved further. From a computational point of view, the main drawback of these methods for addressing DTA on large-scale networks is that they are not parallelizable. This is because all algorithms need to know the last iteration results to determine the next best path

flow for the next iteration. Indeed, they need the state of the network before carrying out the next iteration. Therefore all of the existing methods work behave as serial algorithms to find the UE.

The goal of this study is to overcome the drawbacks of serial algorithms. This study proposes a new solution method based on the Simulated Annealing (SA) method and uses parallel simulations to better explore the solution space. The algorithm is developed generally to solve traffic assignment with parallel computation in order to consider more than one path distribution per iteration. It is obvious that with parallel simulation, the algorithm is going to run more simulations in comparison with existing methods but it is expected to carry out a better exploration of the solution space and consequently achieve better solutions in terms of quality and closeness to the optimal solution. Moreover with parallelizing the framework, the algorithm could solve the problem with better computation time in comparison with classic methods.

2 Methodology

SA algorithm is a meta-heuristic method, it is inspired by annealing in metallurgy. The basic simulated annealing algorithm is presented in [4]. This study redesigns and adapts the classic SA to simulation-based traffic assignment. Figure 1 presents the solution algorithm of this study:

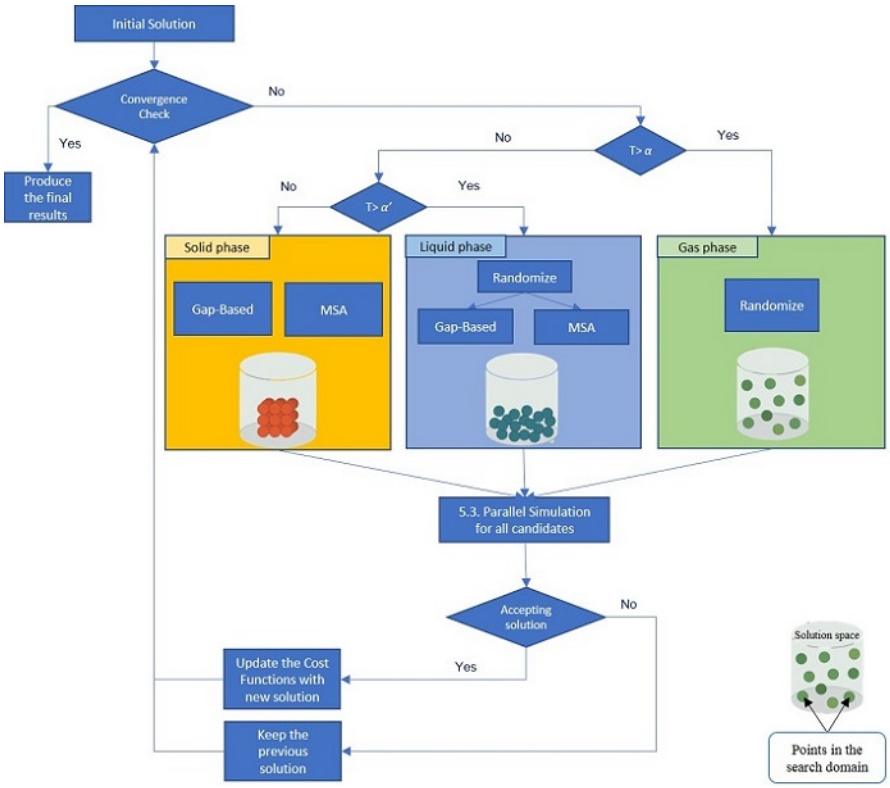


Figure 1: Solution algorithm flowchart

The algorithm starts with an initial solution which is generated randomly. The next solution is generated with respect to the current one based on the temperature (T) of the current iteration. The current phase of the iteration depends on the temperature of the process. Inspired by the physics of matter, this study distinguishes three different methods to generate a neighbor solution, gas, liquid and solid; these methods represent the states of matter in nature. When the temperature is high ($T > \alpha$ where α denotes the boiling temperature), the *gas method* is applied. During the SA algorithm, by decreasing the temperature the algorithm enters the liquid phase ($\alpha > T > \alpha'$ where α' denotes the melting temperature) and then the *liquid method* is applied. When the temperature is quite low ($T < \alpha'$), the *solid method* is applied.

In the gas phase, we explore the solution space without limitation of any step size. Therefore, the candidates for neighbor solution are generated randomly for path flow distribution. The algorithm applies the process on every OD by changing randomly their flow assignment with respect to the constraints applying to the demand (feasible OD-assignment). In the liquid Phase, we target exploring the solutions space randomly and also apply step size methods. First, we apply a randomizing process on the current solution, Then we optimize it by applying the Method of successive Average (MSA) [5] to get the first solution and the Gap-Based method [3] to get the second solution. In the solid phase, we execute the same process as liquid phase but without randomization. It means the two solutions are generated based on the current solution.

Afterward, the algorithm runs parallel simulations to update the network based on new different path distributions that are obtained form the previous step. For a new solution s', the total gap $TGap(s')$ between the users' travel time and the shortest path travel time is calculated and corresponds to the energy of solution (E) compared to the current solution s. The last step consists in making a decision about accepting one of the best new solution based on $TGap$ compared to the current solution of the algorithm. The acceptance decision is made by the binary test. If $P_s \geq R_s$, the new solution will be accepted. Here, $P_s = e^{\frac{-\nabla E}{T}}$, $\nabla E = TGap(s') - TGap(s)$ and R_{tr} is the random number ($0 < R_{tr} < 1$). The quality of the solution is evaluated in the convergence check step. At the end of each iteration the temperature is decreased ($T = \frac{T_0}{ln(k+1)}$ where T_0 denotes the initial temperature and k denotes the iteration index) and the algorithm iterates until converging to the optimal solution or the lowest temperature (T_{min}) is reached.

3 Numerical Experiments

In order to compare the performance of this new method, this study evaluates the algorithm in the static case and compares the method with MSA method and then apply the method to DTA problem for the large-scale test case. In the static case, the method is applied to a 5×5 grid network with static cost functions. The primary results for 3 and 6 ODs with the fixed demand

of 50 users per OD are presented in Figure 2. The results shows that the new method dominates the MSA method even in the small-scale and static network. We are currently running simulations on a large-scale network (Lyon 6e + Villeurbanne: 1,883 Nodes, 5,935 Links, 94 Origins, 227 Destinations, 54,190 trips) with dynamic implementation and the results are very promising.

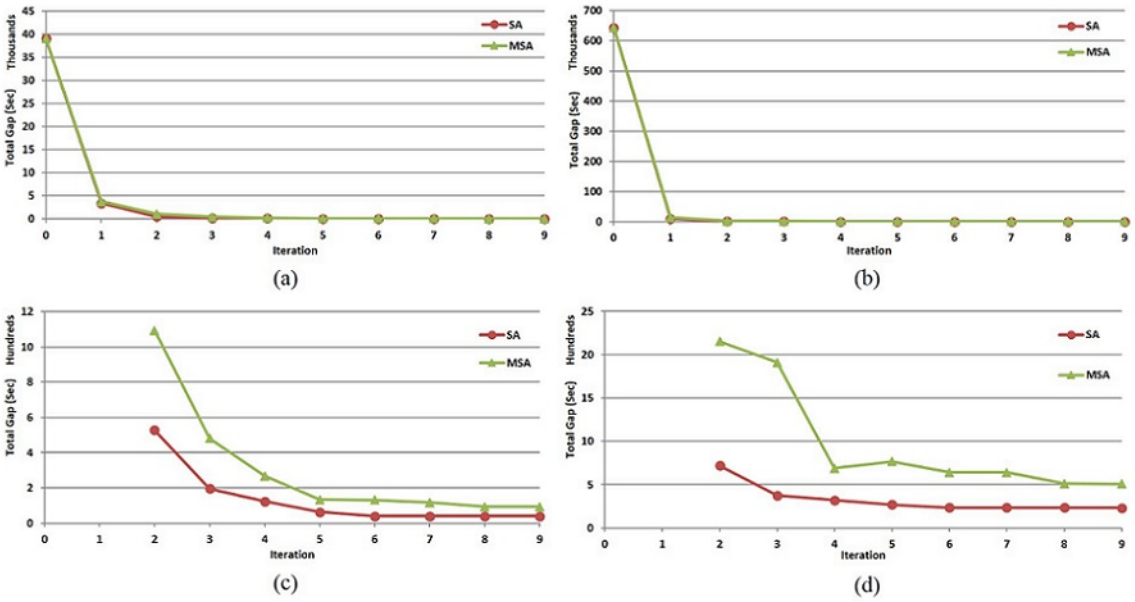


Figure 2: Total gap in 5×5 grid network for SA and MSA methods.
(a), (c): results for 3 ODs. (b), (d): results for 5 ODs.

References

[1] Y. Wang, W. Szeto, K. Han, and T. L. Friesz, "Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications," *Transportation Research Part B: Methodological*, 2018.

[2] W. Szeto and H. K. Lo, "Non-equilibrium dynamic traffic assignment," in *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic TheoryUniversity of Maryland, College Park*, 2005.

[3] C.-C. Lu, H. S. Mahmassani, and X. Zhou, "Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem," *Transportation Research Part B: Methodological*, vol. 43, no. 3, pp. 345–364, 2009.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[5] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

Optimizing Package Express Operations in China

Baris Yildiz

Department of Industrial Engineering
Koc University, Istanbul, Turkey
Email: byildiz@ku.edu.tr

Martin Savelsbergh

H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology, Atlanta/ USA
Email: martin.savelsbergh@isye.gatech.edu

1 Extended Abstract

Package express companies world-wide are facing a rapidly changing environment due to the explosive growth of e-commerce and due to the push by retailers to satisfy their customers' desire for instant gratification by offering faster and faster delivery times. This is especially true in China where the value of B2C e-commerce in 2015 reached $766.5 billion and where more than 430 million people shopped online that year.

To be able to accommodate the expected growth in demand volume and the need for faster service offerings, package express companies are looking for optimization-based tools to support both their planning and operations functions. Motivated by the environment encountered at SF Express, we have designed and implemented optimization models to support express shipment network design which take into account many of the critical features of the environment, such as the integration of ground and air operations, the use of company-owned capacity (in the form of cargo planes) as well as purchased capacity (in the form of belly capacity offered by commercial airlines), the need to offer of multiple service products, and shipments entering the system throughout the day. This environment is far more complex than those typically presented and analyzed in the literature. However, there is a clear practical need for optimization tools that incorporate these complicating features, and, equally important, that can handle instances of the size encountered in

practice. Our efforts are a first step towards providing the industry with such tools and we hope that these efforts stimulate other researchers to do the same.

To accommodate varying demand distributions and densities (due to differences in the markets) we design and implement two optimization models, which can be viewed as being at opposite ends of a spectrum. For the small package market, in which demand is more densely distributed over the service area, and cargo plane capacity is the limiting resource, efficient use of the cargo plane capacity is critical to reducing operational costs. In this case, using cargo planes to serve a limited set of high-volume origin-destination pairs with direct flights is an effective strategy to maximize the utilization of the available transportation capacity, among others because it avoids the use of a "central hub" with time-consuming sorting operations. On the other hand, for the high-value items/products market, in which demand is smaller and less densely (and more evenly) distributed over the service area (compared to the small package market), available flight time is the limiting resource (rather than cargo plane capacity) and the use of transshipment, i.e., allowing planes to meet and exchange cargo at certain locations) is a more effective strategy. In both cases, we formulate and solve novel mixed integer programming models, where, because of the size of real-life instances, we have to exploit the special structure of the models and, in one case, rely on column generation techniques for their solution.

To summarize, the main contributions of this study are the following.

- To accommodate different demand profiles, we propose two novel models to determine high-quality express shipment network designs maximizing company profit.
- Different from can be found in the literature, our models (1) integrate ground and air transport options, (2) consider company-owned and purchased air capacity, (3) consider multiple service products, and (4) consider realistic order arrival patterns. We propose efficient exact algorithms that exploit specific structure to solve large real-life instances.
- We conduct an extensive computational study using real-life data to obtain valuable managerial insights.

Our study differs from the existing literature in several respects. Almost all of the existing studies [5, 2, 4, 1, 3, 6, 7] focus on hub-and-spoke networks, whereas we do not impose any network structure. In our direct shipment model, we only consider transporting shipments from one airport to another, eliminating the need for sorting and repackaging at a hub airport. In our transshipment model, we consider transshipments, i.e., some locations function as "hubs" where planes meet and exchange their cargo, but different from more traditional hub-and-spoke systems, no sorting takes place at the the transshipment hubs. Another important difference is our treatment of ground transportation. In most previous studies, the assignment of stations (cities without airports) to gateway hubs (cities with airports) are fixed. In our models, such assignments are an integral

part of the service network design. Contrary to most traditional express shipment service network design settings, our models consider both company-owned capacity as well as purchased capacity (belly capacity available on commercial flights), which is especially important in China. Finally, we consider an environment in which the express carrier offers multiple service classes, and in which shipments enter the air transportation network throughout the day.

1.1 Direct Shipments Model

In the direct shipments model DSM, a demand that is transported by air only occupies one flight leg. That is, in DSM, a demand is either transported from its origin city to its destination city by ground transportation, or it is transported in three phases: (1) from its origin city to an airport city using ground transportation, (2) from an airport city to another airport city on a single flight, either on a company-owned cargo plane or on a commercial plane (i.e., using purchased belly capacity), and (3) from an airport city to its destination city using ground transportation. In the latter case, one or both of the ground transportation phases may be "empty". The DSM makes the most effective use of the company-owned cargo plane capacity, as any demand transported using the cargo planes uses the capacity for the shortest possible distance.

For the direct shipments, we have developed an optimization model that is capable of handling realistic-size data. More specifically, we have use it solve an instance with more than 100,000 demands, 6,000 commercial flights (from 86 origins to 109 destinations), 34 hub-cities, 56 cargo planes (with three different types), and more than 2,500 city pairs for the ground transportation. The spatial distribution of the demand is illustrated in Figure 1 and the schedules (for the three types of cargo planes) found by our solution approach can be seen in Figure 2.

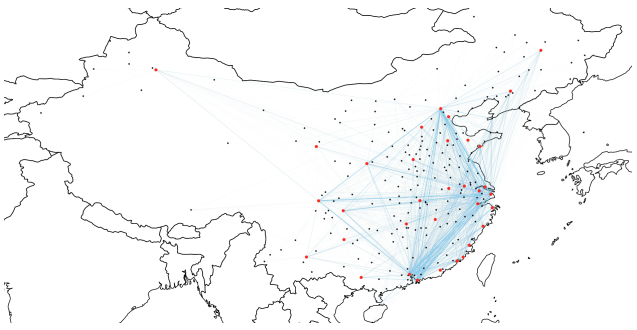
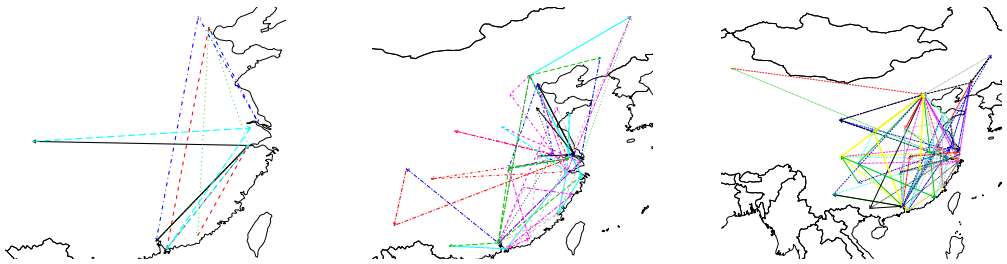


Figure 1: The spatial distribution of express package demand.

1.2 Transshipment Model

In the transshipment model (TSM), a demand transported by air may occupy more than one flight leg, often, but not necessarily, on more than one cargo plane. That is, cargo planes are allowed to



(a) Type 1 planes (42 tons) (b) Type 2 planes (28 tons) (c) Type 3 planes (14 tons)

Figure 2: The cargo plane routes found by the optimization algorithm.

meet in a transshipment location to exchange some or all of their cargo. Conceptually, each demand uses one pickup and one delivery flight that meet at a transshipment location. If the transshipment location happens to be the hub-city where the demand enters or exits the air network, then the demand uses only a single flight (either a pickup or a delivery flight). In TSM, each cargo plane's route is composed of two distinct parts: a pickup flight and a delivery flight (where one of them may be the "empty" flight). On the pickup flight, the cargo plane collects demands at various hub-cities to take them to a transshipment location. On the delivery flight, the cargo plane takes demands from the transshipment location and drops them of at various hub-cities. As in the direct shipment model, ground transportation can be used to transfer demand from/to a non-hub city to/from a hub-city. The TSM makes most effective use of the flying time available for company-owned cargo planes to provide broad coverage, i.e., seeks to provide connections between a large number of cities (possibly at the expense of reduced capacity utilization).

For the transshipments model, we have developed an optimization approach that utilizes a path segment formulation approach to directly solve realistic-size problems with 100 origin destination pairs (derived from the real world data) and 34 hub-cities all of which can function as a transshipment point. The spatial distribution of the demand is illustrated in Figure 3 and the amount of demand (value and count) that could be covered by various number of cargo planes is shown in Figure 4.

References

[1] A. P. Armacost, C. Barnhart, and K. A. Ware. Composite variable formulations for express shipment service network design. *Transportation science*, 36(1):1–20, 2002.

[2] C. Barnhart and R. R. Schneur. Air network design for express shipment service. *Operations Research*, 44(6):852–863, 1996.

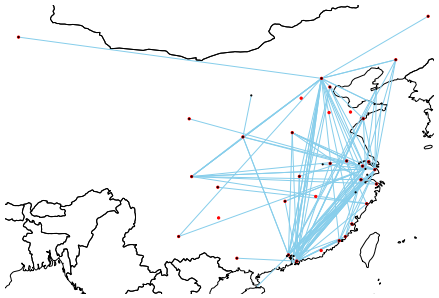


Figure 3: The spatial distribution of high-value items/products demand.

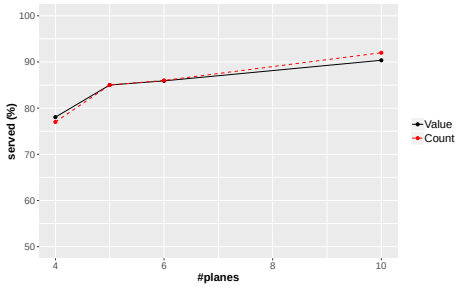


Figure 4: Impact of increasing fleet size.

[3] H. Fleuren, C. Goossens, M. Hendriks, M.-C. Lombard, I. Meuffels, and J. Poppelaars. Supply chain–wide optimization at tnt express. *Interfaces*, 43(1):5–20, 2013.

[4] D. Kim, C. Barnhart, K. Ware, and G. Reinhardt. Multimodal express package delivery: A service network design application. *Transportation Science*, 33(4):391–407, 1999.

[5] M. J. Kuby and R. G. Gray. The hub network design problem with stopovers and feeders: The case of federal express. *Transportation Research Part A: Policy and Practice*, 27(1):1–12, 1993.

[6] I. Louwerse, J. Mijnarends, I. Meuffels, D. Huisman, and H. Fleuren. Scheduling movements in the network of an express service provider. *Flexible Services and Manufacturing Journal*, 26(4):565–584, 2014.

[7] J. M. Quesada Perez, J.-S. Tancrez, and J.-C. Lange. " a multi-hub express shipment service network design model with flexible hub assignment. In *EURO 2016: 28th European Conference on Operational Research*, 2016.

A many-to-many stable matching cost allocation model for multimodal Mobility-as-a-Service

Saeid Rasulkhani, Theodoros P. Pantelidis, Joseph Y. J. Chow*
Department of Civil & Urban Engineering
Tandon School of Engineering
New York University
*Corresponding author's e-mail: joseph.chow@nyu.edu

1 Background

With the increasing ubiquity of multiple forms of "Mobility as a Service" (MaaS) options to travelers provided by both public agencies and private operators, travel forecast models for different transportation network designs need to focus on both the decisions of travelers and operators *[1]*. We need to consider assignment of both flows and cost allocations to users and operators as a descriptive travel forecast model. The problem of determining cost allocation and corresponding stable matches with transferable utility between players is called an assignment game *[2]*, which involves a set of buyers and sellers. Different types of assignment games exist: one-to-one games involve matching individual buyers to individual sellers; many-to-many games match one seller to many buyers, and each buyer can themselves be matched to many sellers *[3]*.

Applications of matching in multicommodity flow problems can be found in the network literature. However, much of this literature either looks only at coalition formation between operators ignoring the allocations to decentralized users (e.g. *[4]*) or propose specific cost allocation mechanisms between users and operators (e.g. *[5]*-*[6]*). Neither address the problem of assigning travelers onto an operator route composed of a sequence of nodes with line capacities and route-level cost allocation decisions of operators.

Rasulkhani and Chow *[7]* proposed a many-to-one assignment game in which users constrained by line capacities on routes are each matched with one operator of a bundle of routes to get from an origin to a destination (OD). The output of the model is a set of unimodal route flows for travelers under line capacity constraints with the corresponding stable outcome space for cost allocations based on the core.

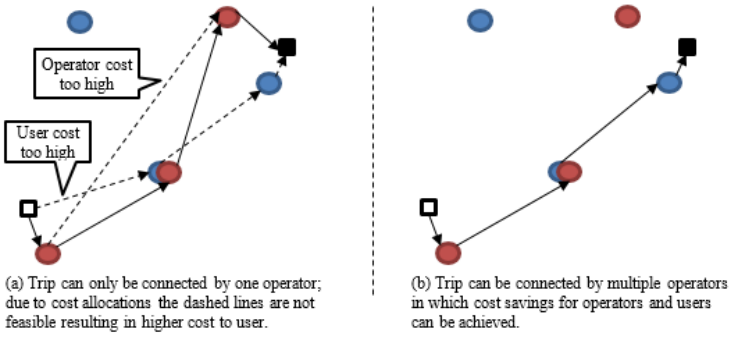


Fig. 1. Illustration of difference in methodology between (a) *[7]* and (b) this study.

We propose to extend that work in a significant new direction by considering many-to-many matches between users and operators. In effect, a single traveler's trip may be split into multiple legs served by different operators to get them to the destination while each operator serves multiple users up to a certain line capacity. The difference from *[7]* is illustrated in Fig. 1. The output of

such a model is not just the flows, but the range of cost allocations needed to incentivize the users and operators to accept those flows.

2 Methodology

The multimodal assignment game is a multicommodity capacitated fixed charge network design problem (MCND) shown in Eq. (1) – (4). Let $G = (N, A)$ be a directed network, where N is the set of nodes and A set of links. We define t_{ij} as the user's travel cost on link $(i,j) \in A$. We also define c_{ij} as the cost of operation of that link and w_{ij} as the capacity of the link. $N_i(+)$ and $N_i(-)$ respectively are the sets of incoming and outgoing nodes from node $i \in N$ in the network. Flow on link $(i,j) \in A$ for each user $s \in S$ is x_{ij}^s, where user s is characterized by demand d^s for an OD pair. A binary variable y_{ij} indicates if a link $(i,j) \in A$ is operated. The MCND is well-defined in the literature. The problem can be solved using conventional MCND methods like the branch-and-price-and-cut algorithm.

$$\min \sum_{(i,j)\in A} \sum_{s\in S} t_{ij} x_{ij}^s + \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{1}$$

$$s.t. \sum_{j\in N_i(+)} x_{ij}^s - \sum_{j\in N_i(-)} x_{ji}^s = \begin{cases} d^s & if\ i = O(s) \\ -d^s & if\ i = D(s) \\ 0 & otherwise \end{cases} \qquad \forall i \in N, s \in S \tag{2}$$

$$\sum_{s\in S} x_{ij}^s \le w_{ij} y_{ij} \qquad \forall (i,j) \in A \tag{3}$$

$$x_{ij}^s: integer \quad y_a: binary \tag{4}$$

Let R be the set of user paths for user $s \in S$. $A_r \subseteq A$ is the set of links of path $r \in R$. When a user is assigned to a path, a payoff is generated and divided between the user and operator(s) of the links of that path. Each operator $f \in F$ stays in the coalition if they get an allocation greater than or equal to the amount they can earn by unilaterally breaking away from the coalition and making another coalition. Let $C(s,x)$ equal a set of links that user s is matched to under link assignment x from Eq. (1) – (4). Let $L(f)$ be the set of links owned by operator $f \in F$. We denote p_{ij}^s as the ticket price that each of the individual user s should pay to link $(i,j) \in A$. The ticket price that operator $f \in F$ gets from the users that are matched to his links, should cover the operation cost of the links f is operating. Moreover, U_s is the utility that user s gets from completing their trip.

Feasible outcome: the outcome $\big((u,p);x\big)$ is feasible if:

(i) $u_s + \sum_{(i,j)\in A_r} p_{ij}^s = U_s - \sum_{(i,j)\in A_r}(t_{ij})\ if\ x_{ij}^s \ge 1\ and\ u_s \ge 0,\ p_{ij}^s \ge 0\ \forall s \in S,\ (i,j) \in A$

(ii) $\sum_{(i,j)\in L(f)} \sum_{s\in S} p_{ij}^s\, x_{ij}^s \ge \sum_{(i,j)\in L(f)} C_{ij} \quad \forall f \in F$

Let μ_{ij} be the revenue loss of link $(i,j) \in A$ when the operator of that link wants to make a coalition with a new user, where $\mu_{ij} = \min_s \{p_{ij}^s \mid (i,j) \in A_r\}$ if $\sum_{s\in S} x_{ij}^s = w_{ij}$, $\mu_{ij} = 0$ if $\sum_{s\in S} x_{ij}^s < w_{ij}$, and $\mu_{ij} = c_{ij}$ if $\sum_{s\in S} x_{ij}^s = 0$.

Stable outcome: The feasible outcome $\big((u,p);x\big)$ is stable if Eq. (5) is satisfied.

$$\sum_{(i,j)\in A_r} \mu_{ij} + u_s \geq \delta_{sr}\left[U_s - \sum_{(i,j)\in A_r}(t_{ij})\right] \qquad \begin{matrix}\forall r \notin C(s,x)\\ \forall s \in S\end{matrix} \qquad (5)$$

We divide the systems that we are analyzing into two different categories, centralized and decentralized decision-making systems. In centralized decision making, each operator can own more than one link in the network, whereas decentralized system operators own only one link (i.e. each link is an operator).

In a decentralized system, we propose to obtain the allocation without enumeration by perturbing allocations to ensure the resulting paths for each OD pair matches the costs of positive link flow outcomes in the MCND using inverse optimization (see *[8]*) in a process similar to *[4]*. Algorithm 1 is proposed to create this stable outcome space.

Algorithm 1: Construct stable outcome space

For each user s,

Step 0. Update the travel cost $c_{ij}^s = t_{ij} + p_{ij}^s$ for each link $(i,j) \in A_{r\in C(s,x)}$
For links $(i,j) \in A_{r\notin C(s,x)}$:
If $\sum_{s\in S} x_{ij}^s = w_{ij}$ then create w_{ij} copies of link (i,j) and assign each with cost $c_{ij}^s = t_{ij} + p_{ij}^s$;
If $0 < \sum_{s\in S} x_{ij}^s < w_{ij}$ then leave the link (i,j) with the cost $c_{ij}^s = t_{ij}$;
If $\sum_{s\in S} x_{ij}^s = 0$ then update the link (i,j) cost as $c_{ij}^s = t_{ij} + C_{ij}$.

Step 1. Add the following to the constraint set:
If $(i,j) \in A_{r\in C(s,x)}$ then $c_{ij}^s - (\pi_i^s - \pi_j^s) = 0$, where π_i^s is the node potential for user s at node i;
If $(i,j) \in A_{r\notin C(s,x)}$ then $c_{ij}^s - (\pi_i^s - \pi_j^s) \geq 0$.

Step 2. Make the constraints to address feasibility condition (ii) by setting $\sum_{s\in S} x_{ij}^s\, p_{ij}^s \geq c_{ij}$ for each link $(i,j) \in A$.

The cost allocation model is to maximize an objective (e.g. set prices to obtain user- or operator-optimal prices), subject to the stability condition which is either Eq. (6) or the constraints constructed in Algorithm 1 in decentralized system. For the constraints that are made by Algorithm 1, we use a Dantzig-Wolfe decomposition method to solve the cost allocation model since the dual of the inverse optimization problem has a "primal block angular" structure.

3 Experiments

We consider a 4-node network first, as shown in Fig. 2. The numbers on the links represent travel times. We assume 6 O-D pairs $s = \{12,13,23,32,41,42\}$, where demand for (1,2) is 4 and the rest of the OD pairs hold unit demand: $d_s = [4,1,1,1,1,1]^T$. Each pair s has a utility of completing the journey $U_s = 20$ and there is a capacity of $w_{ij} = 2$ for each link. In centralized system, the red, green and black link colors in Fig. 2 represent different operators.

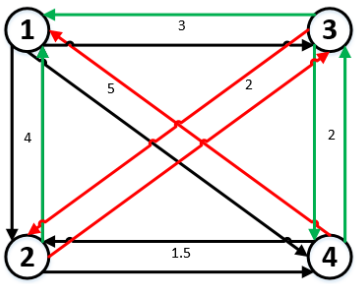


Fig. 2. Each color represents a different transit operator.

The solution flow and cost allocation space (which ranges between the user-optimal and operator-optimal solutions) for both centralized and decentralized systems are shown in Table 1. The observed flows and stable payoff range required from the users is the result of link capacities and the combination of operator network interactions with each other. The centralized matching for user (1,2) includes both

red and black operators. Each of the passengers of this OD would pay between \$4.25/person to \$11/person in which red operator would earn between \$1.25/person to \$3.125/person. The cost for the red operator to leave the coalition is between \$0 to \$7.5 and between \$2 to \$21.5 for the black operator. The stable payoff space is wider under the centralized system.

Table 1. Model results

	Operated links	12	13	14	32	42	13	23	32	41	42	Total
	O-D	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)	(1,3)	(2,3)	(3,2)	(4,1)	(4,2)	
	Total flow x_{ij}	2	2	1	2	2	2	1	2	1	2	
Decentralized	p_{ij}^s **User Optimal**	9	4	5	2	0	1	5	3	5	5	39
	p_{ij}^s **Operator Optimal**	17	2.5	5	11.5	8	7.5	11.5	14.5	10.5	18	106
Centralized	Operator											
	p_{ij}^s **User Optimal**	8	0	0	5	4	0	3.5	6.5	0	0	27
	p_{ij}^s **Operator Optimal**	17	1.5	0	12.5	13	16	11.5	18	10.5	18	118

In addition to the toy network, the proposed model and algorithm are tested on two case studies. The first one deals with evaluating flow and range of cost allocations for a hub-and-spoke transit system with feeder buses and a trunk metro line. We demonstrate how to apply our model to evaluate the sensitivity of the feeder buses' negotiating power against the metro relative to vehicle capacity, demand patterns, travel costs, and consolidation of feeders. In a second study, we determine prices that can be charged for an operating strategy *[9]* in which ride-share service borrows from public transit capacity in delivering passengers.

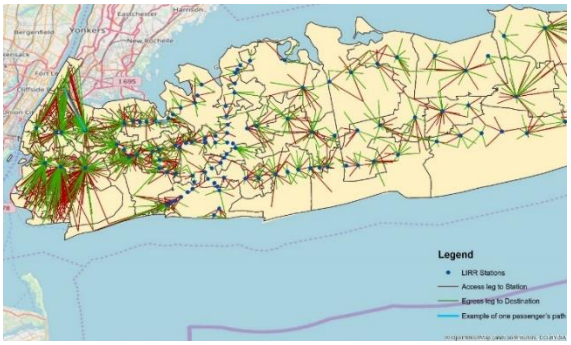


Fig. 3. One of the case study networks.

References

[1]. Djavadian, S., Chow, J. Y. J., 2017. An agent-based day-to-day adjustment process for modeling 'Mobility as a Service' for a two-sided flexible transport market. *Trans. Res. Part B* 104, 36-57.

[2]. Shapley, L.S., Shubik, M., 1971. The assignment game I: the core. *Int. J. Game Theory*, 1(1), 111-130.

[3]. Roth, A. E., Sotomayor, M. A. O., 1990. *Two-sided matching: a study in game-theoretic modeling and analysis*, Cambridge University Press, New York.

[4]. Agarwal, R. and Ergun, Ö., 2008. Mechanism design for a multicommodity flow game in service network alliances. *Operations Research Letters*, *36*(5), pp.520-524.

[5]. Rosenthal, E. C., 2017. A cooperative game approach to cost allocation in a rapid-transit network. *Trans. Res. Part B* 97, 64-77.

[6]. Wang, X., Agatz, N., Erera, A., 2017. Stable matching for dynamic ride-sharing systems. *Trans. Sci.*, in press, doi: 10.1287/trsc.2017.0768.

[7]. Rasulkhani, S. and Chow, J. Y. J., 2017. Route-cost-assignment with joint user and operator behavior as a many-to-one stable matching assignment game. *arXiv preprint arXiv:1711.11230*.

[8]. Xu, S. J., Nourinejad, M. Chow, J. Y. J., and Lai, X., 2018. Network learning via multi-agent inverse transportation problems. *Trans. Sci.*, in press, doi: 10.1287/trsc.2017.0805.

[9]. Ma, T.Y., Chow, J.Y.J. and Rasulkhani, S., 2018. An integrated dynamic ridesharing dispatch and idle vehicle repositioning strategy on a bimodal transport network. *Proceedings of 7th Transport Research Arena TRA*.

A strategic Markovian traffic equilibrium model for capacitated networks

Maëlle Zimmermann
Department of Computer Science and Operations Research
University of Montreal, Canada
Email: maelle.zimmermann@umontreal.ca

Emma Frejinger
Department of Computer Science and Operations Research
University of Montreal, Canada

Patrice Marcotte
Department of Computer Science and Operations Research
University of Montreal, Canada

1 Introduction

Traffic equilibrium models are fundamental tools for the analysis of transportation networks performance as well as their design and planning. The traffic assignment problem consists in predicting arc flows over a network, given the known travel demand for each origin-destination (OD) pair. Flows are determined by the interaction of two mechanisms, users' travel decisions and congestion. Users' route choice preferences are incorporated in a generalized travel cost function which individual travelers aim to minimize, the primary component of which being travel time. Congestion is generally modeled by letting travel impedance functions depend on the usage of the network. As path costs increase with the amount of flow, travelers are induced to reroute on cheaper, less congested paths. The equilibrium assignment of travelers to routes is thus the result of a fixed point problem which is usually solved in an iterative manner. However, the classical equilibrium principles do not hold any more when side constraints, such as arc capacities, are entered into the model. A solution to that issue, proposed in [1], is to embed within the users' objective function the probability that a link be unavailable, thus introducing a stochastic element that induces the strategic behaviour of users.

The main contribution of this work is to generalize this model by including another source of

stochasticity, stemming from users' imperfect knowledge of travel times. By adopting the framework of Markovian equilibrium introduced by [2], our model then generalizes the latter by incorporating arc capacities. More specifically, we embed the concept of strategies governing travelers' movements under capacity constraints in a Markovian traffic equilibrium setting. The key paradigm is to view strategies with recourse, according to which travelers readjust their path when reaching a saturated arc, as route choice behavior under imperfect information, similarly to [3]. In order to deal with partial information, we expand the state space of the Markov Chain in [2], such that a state encompasses two variables, an arc and an information set. User path choice behavior is then characterized by sequences of local arc choices depending on the current state and the destination. The network loading gives rise to availability probabilities, which are akin to access probabilities in [1] and at the same time characterize action-state transition probabilities in the context of Markovian decision processes. The strength of the approach is to encompass two sources of stochasticity in the model by incorporating both unobserved elements and the risk of failure to access an arc in the cost of travel. In addition, the proposed model is arc-based and does not require path enumeration.

2 Strategic Markovian traffic equilibrium model

We consider a directed connected graph $G = (\mathcal{A}, \mathcal{V})$, where $\mathcal{A}$ is the set of arcs, or links, and $\mathcal{V}$ is the set of nodes. Links are denoted either by k or a and $\mathcal{A}(k)$ is the set of outgoing links from the tail node of k. We assume that every link a has a strict capacity u_a and an associated generalized cost c_a. Assuming users have an imperfect knowledge of costs, we model *perceived* arc costs as random variables $\widetilde{c}_a = c_a + \mu\epsilon_a$, letting the measured arc cost be disrupted by an error term with $E(\epsilon_a) = 0$. We add absorbing links without successors to each destination node and call $\mathcal{D}$ the set of destination links. We consider the demand to originate from each network link, and let g^d characterize the vector of demand from each link given a destination $d \in \mathcal{D}$. We assume that the network has sufficient capacity to accommodate the whole demand.

Users traveling in this network aim at finding the shortest path to their destination $d \in \mathcal{D}$. However, because of limited network capacity, some arcs may be saturated and thus inaccessible depending on route choices made by other travelers. Similarly to [1], we assume a realistic modeling of user behavior, dictating that travel decisions be strategic and include recourse actions, should a link in the preferred itinerary turn out to be unavailable. In addition, we make the hypothesis that travelers do not know in advance what arc will prove to be available, and only observe the outcome when reaching the source node of an arc. Under these assumptions, the problem bears similarities to the stochastic shortest path problem in a probabilistic network. As observed in [3], stochastic programming with recourse can be viewed as a stochastic control problem with imperfect

information, and may be solved with dynamic programming methodology. Namely, instead of defining recourse actions, user behavior may equivalently be characterized by an optimal policy given the current state, where the state indicates the realization of the random variables. Below, we explain how we formulate the model following this paradigm.

We assume that the set of available outgoing arcs from link k is a random subset of $\mathcal{A}(k)$, and define the random vector I_k, which indicates whether each outgoing arc is accessible and may take values in $\Omega_k = \{0,1\}^{|\mathcal{A}(k)|}$. Consequently, we define a state (k, i_k) as a set of two variables, i.e., a link k and a realization i_k of random vector I_k. The set of states at link k is denoted $\mathcal{S}_k$, while the set of all possible states is denoted $\mathcal{S}$. A policy, or action, is then a choice of outgoing arc among the set $\mathcal{A}(s)$ of available links depending on the current state $s = (k, i_k)$. For unvisited arcs k, the random vector I_k follows availability probability distribution π_k, which has support on $\{0,1\}^{|\mathcal{A}(k)|}$. Upon arrival at the tail node of arc k, the user learns the realization of I_k. Therefore, travelers choose their paths sequentially in a dynamic fashion, choosing in each state an action that leads stochastically to a new state.

Travelers' route choice behavior is characterized by the destination specific choice probability matrix $P^d = \{P^d_{s,a}\}_{s\in\mathcal{S}, a\in\mathcal{A}}$, which describes in what proportion individuals choose each action conditionally on the state and the destination. The role of availability probabilities π is analog to that of state transition probabilities conditional on choices in a Markov Decision Process. Given a state $s_t = (k, i_k)$ and an action $a \in \mathcal{A}(s)$, the probability $\Pr(s_{t+1}|s_t, a)$ of reaching the new state $s_{t+1} = (a, i_a)$ is given by the distribution π_a of random vector I_a. In other words, the new state consists of the chosen available link and a realization of the availability random vector at that link. We can here draw a parallel with the model of [2], where the choice of outgoing link may also be viewed as a choice of action leading to a new state.

We can formulate the equilibrium problem as a variational inequality. We first define the expected cost w^d_a of actions $a \in \mathcal{A}(s)$ as

$$w^d_a = c_a + E_{i_a\sim\pi_a} V^d(a, i_a), \tag{1}$$

and the expected minimum cost of traveling to destination d from state (k, i_k) by the Bellman equation

$$V^d(k, i_k) = E_{\epsilon_a}\left[\min_{a\in\mathcal{A}_i(k)} \left\{c_a + E_{i_a\sim\pi_a} V^d(a, i_a) + \mu\epsilon_a\right\}\right]. \tag{2}$$

We then define the cost $C^d_{s,a}$ as the sum $w^d_{s,a} + \mu \ln(P^d_{s,a})$, where $w^d_{s,a}$ is equal to w^d_a if $a \in \mathcal{A}(s)$ and ∞ otherwise. Then, for each destination, the equilibrium choice probabilities $P^*_{s,a}$ are the solution of

$$\langle C_s(P^*), P^*_s - P_s\rangle \leq 0 \quad \forall P \in \mathcal{P},\ \forall s \in \mathcal{S}, \tag{3}$$

where the destination index is omitted for the sake of simplicity.

		Expected minimum cost V_o^d				Gap (%)
Heuristic	μ	**OD1**	**OD2**	**OD3**	**OD4**	$g_R(P)$
Common step size	0.5	119.74	138.92	114.72	99.61	$2.15 \cdot 10^{-2}$
Disaggregate step size		119.69	138.84	114.15	99.60	$3.21 \cdot 10^{-2}$
Common step size	5	116.83	131.42	113.00	96.23	$6.26 \cdot 10^{-3}$
Disaggregate step size		117.05	131.34	112.61	96.16	$4.13 \cdot 10^{-2}$
Common step size	10	112.00	119.25	107.38	88.50	$7.99 \cdot 10^{-2}$
Disaggregate step size		112.59	119.80	107.21	88.57	$1.10 \cdot 10^{-1}$
Common step size	20	95.57	106.02	94.11	80.25	$9.10 \cdot 10^{-3}$
Disaggregate step size		95.18	105.59	93.38	79.88	$7.62 \cdot 10^{-2}$

Table 1: Expected minimum cost of OD pairs after 1000 iterations of the solution algorithm with different values of μ

3 Results

The main challenge to solving the problem is that the cost $C_{s,a}^d$ of state-action pairs in (3) is not available in closed form as a function of choice probabilities P^d. Instead it depends on P^d through availability probabilities π, which induce nonlinearity in the cost mapping and must be obtained through an inner algorithm related to that found in [1].

We apply the proposed model to several networks, in particular the Sioux Falls network depicted in [1]. We use the method of successive averages and two inner algorithms to find the equilibrium flows and choice probabilities in the network. We resort to a well-defined gap function to evaluate the proximity of the iterate with the equilibrium solution. We compute the equilibrium for several values of μ, the scale of the random term ϵ and display the results in Table 1. We find that as the value of μ tends to zero, the solution is close to a deterministic user equilibrium (with respect to the arc costs), whereas when μ becomes large it is equivalent to a random walk on the network.

We conclude by emphasizing that the following work has very recently been submitted for publication to Transportation Science, but has not been presented before at an international conference.

References

[1] Marcotte, P., Nguyen, S., and Schoeb, A. "A strategic flow model of traffic assignment in static capacitated networks", *Operations Research*, 52(2):191212, 2004.

[2] Baillon, J.-B. and Cominetti, R. Markovian traffic equilibrium, *Mathematical Programming*, 111(1-2):3356, 2008.

[3] Polychronopoulos, G. H. and Tsitsiklis, J. N. "Stochastic shortest path problems with recourse", *Networks: An International Journal*, 27(2):133143, 1996.

Data-Driven Transit Network Design at Scale

Dimitris Bertsimas
Sloan School of Management and Operations Research Center
Massachusetts Institute of Technology

Yee Sian Ng
Operations Research Center
Massachusetts Institute of Technology

Julia Yan
Operations Research Center
Massachusetts Institute of Technology, 77 Massachusetts Avenue
Email: jyyan@mit.edu

1 Introduction

Public transit is crucial for sustainability, efficiency, and equity in serving urban populations. However, it faces significant outside competition from ride-sharing companies and private bus or shuttle services. Many cities such as Philadelphia, Los Angeles, and Washington D.C. are seeing declining bus ridership, prompting transit authorities to consider what can be done to halt this decline. A recent bus network re-design in Houston led to a 6.8% increase in ridership across the bus and light rail networks [1], inspiring cities such as Boston, Philadelphia, and St. Louis to also consider re-designing their bus networks.

Our goal will be to optimally design a transit network in order to maximize ridership subject to budget constraints, and we aim to do so both realistically and tractably. In order to model ridership, we will have to account for the key aspects of the commuter decision-making process, which involves both travel time and transfers. We also scale our models up to a 410-station, 4,893 edge network taken from Boston using column-and-constraint generation.

Much of the early work on transit network design focused on heuristic solution methods [2, 3, 4]. Typically, the origin-destination demand matrix was sorted from highest to lowest demand, and bus routes were generated using fast shortest-path computations between high-demand nodes. generate an initial line set by computing the shortest paths between terminal nodes, and then

uses local search to iteratively improve the total travel time on the network. Along a similar vein, metaheuristics such as genetic algorithms [5, 6], simulated annealing [7], and tabu search [8] have also been used to iteratively improve upon initial heuristically-generated route sets.

Another area of work has employed mathematical optimization to solve network design problems. The benefit of mathematical optimization is a certificate of optimality; however, many models have had scalability issues at practical network sizes. Many papers restricted their attention to the optimal selection of a subset of bus stops [9] or heuristically-generated bus routes [10, 11], without considering the generation of new routes. Even with these limitations, they only scaled to networks of tens of stations.

Relatively fewer papers have addressed exact route generation due to further scalability issues. Approaches based on mixed-integer optimization [12, 13] and constraint programming [14] have scaled only to tens of stations. In contrast to these smaller-scale examples, [15] employed column generation to scale up their model to a network of hundreds of stations and one thousand edges, a truly large-scale application. However, they remained closely tethered to the original network design by only considering edges that already existed in the network in their computational study, so that the new lines that were produced were rearrangements of existing lines.

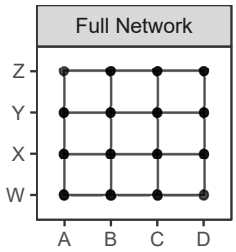
Our approach builds on this well-established framework of beginning with an initial set of lines, then selecting the best subset of these lines to operate, and finally iteratively generating new candidate lines. We use column-and-constraint generation to provide optimality guarantees, and scale to a network of hundreds of stations and thousands of edges.

2 Methods

We formulate an integer optimization model that addresses maximization of ridership, budget constraints, and a model of commuter choice. Our commuter choice model assumes that commuters will take public transit if they can get from origin to destination with at most one transfer; later in this section, we will also build in travel times. Although our model can generalize to arbitrary number of transfers, we restrict our attention to single transfers so as not to place undue burdens on commuters. At its core, our model is a best subset selection problem: similar to many approaches in the literature, we assume that the problem is initialized with some set of candidate bus lines, and the transit agency's decision is to decide which bus lines to operate.

The immediate drawback with the approach of initializing with a set of candidate bus lines is that such a set must be exhaustive in order to guarantee the quality of the solution. Rather than starting with all possible bus lines, we use column-and-constraint generation to selectively generate bus lines, focusing only on those that are profitable for our problem. For an overview of column-and-constraint generation, see [16]. We show that the profitability condition for a new

Figure 1: Synthetic bus network generated from single-transfer model with travel time constraints



bus line has a simple and intuitive interpretation, which informally is that the increase in ridership must outweigh its costs. Bus lines satisfying this profitability condition can be generated using integer optimization.

A limitation of the model thus stated is that one of its solutions would be to connect all stations in a Hamiltonian path, if such a path is within budget. However, this solution is clearly not appealing for commuters, particularly those between terminal stations, for whom this solution is inefficient due to high travel times. We address travel times by enforcing that for all pairs of stations on a bus line, the travel time between those stations should not be more than some constant factor above the shortest possible travel time between those stations. This is implemented using lazy constraints in our bus line generation integer optimization model. Enforcing this condition is admittedly more stringent than allowing longer bus lines that some commuters may elect not to take. However, we show that this stronger condition is more tractable, resulting in the addition of significantly fewer lazy constraints. Furthermore, we have found in computational experiments on the Boston network that the vast majority of bus lines adhere to this stronger condition, indicating that it is a desirable property in practice.

3 Computational Results

We present a selection of our computational results on both synthetic and real data.

Our synthetic network was a four-by-four grid of stations with equal demand between all pairs of stations. On this problem, our model terminated in eight iterations totaling approximately one minute, producing the network shown in Figure 1. From Figure 1, we see the intuitive appeal of grid networks: every origin can reach every destination with at most one transfer, and travel times remain modest.

We also demonstrate our algorithm's tractability on a real dataset from Boston, comprising 410 stations and 4,893 edges, where any edge was considered if it had length of one mile or less. Demand data was obtained from the Massachusetts Bay Transit Authority (MBTA). Select results for a

range of budgets are shown in Table 1, showing increases in ridership of about 5-15%. Furthermore, our algorithm was tractable, terminating in about 12 hours for each case. These running times are reasonable given that network design is an offline problem, only undertaken once every several years.

Budget	**Original Network**		**Optimized Network**		**Running Time (hrs)**
	Objective	% Ridership	Objective	% Ridership	
50	73,575	70.4%	80,267	76.8%	12.1
100	88,716	84.9%	98,487	94.3%	11.1
150	89,472	85.7%	103,092	98.7%	12.1

Table 1: Objectives values of the original Boston network and the optimized network, and algorithm running times

4 Summary

We have addressed the problem of designing bus lines for urban transit networks. In particular, we seek to maximize ridership on a bus network, accounting for the fact that passengers will choose to take the bus if one of their possible routes is appealing in travel time and number of transfers. In our computational experiments, we demonstrate that our algorithm produces intuitive results on a synthetic network, and demonstrate significant potential gains on a real dataset from Boston comprising hundreds of stations and thousands of edges. All of these are achieved with reasonable running times. This presents opportunities for transit authorities to perform holistic redesign of their transit networks in order to offer a service that is both cost-efficient and appealing to commuters.

References

[1] L. Binkovitz, "A year after bus redesign, METRO Houston ridership is up", Kinder Institute for Urban Research, August 2016.

[2] C.E. Mandl, "Evaluation and Optimization of Urban Public Transportation Networks", *European Journal of Operational Research* 5(6), 396-404 (1980).

[3] A. Ceder and N.H.M. Wilson, "Bus Network Design", *Transportation Research Part B: Methodological* 20(4), 331-344 (1986).

[4] M.H. Baaj and H.S. Mahmassani, "Hybrid Route Generation Heuristic Algorithm for the Design of Transit Networks", *Transportation Research Part C: Emerging Technologies* 3(1), 31-50 (1995).

[5] E. Cipriani, S. Gori, and M. Petrelli, "Transit Network Design: A Procedure and an Application to a Large Urban Area", *Transportation Research Part C: Emerging Technologies* 20(1), 3-14 (2012).

[6] J.L. Walteros, A.L. Medaglia, and G. Riano, "Hybrid Algorithm for Route Design on Bus Rapid Transit Systems", *Transportation Science* 49(1), 66-84 (2013).

[7] F. Zhao and X. Zeng, "Simulated Annealing Genetic Algorithm for Transit Network Optimization", *Journal of Computing in Civil Engineering* 20(1), 57-68 (2006).

[8] N.E. Lownes and R.B. Machemehl, "Exact and Heuristic Methods for Public Transit Circulator Design", *Transportation Research Part B: Methodological* 44(2), 309-318 (2010).

[9] A.T. Murray, "A Coverage Model for Improving Public Transit System Accessibility and Expanding Access", *Annals of Operations Research* 123(1-4), 143-156 (2003).

[10] J.F. Guan, H. Yang, and S.C. Wirasinghe, "Simultaneous Optimization of Transit Line Configuration and Passenger Line Assignment", *Transportation Research Part B: Methodological* 40(10), 885-902 (2006).

[11] H. Cancela, A. Mauttone, and M.E. Urquhart, "Mathematical Programming Formulations for Transit Network Design", *Transportation Research Part B: Methodological* 77, 17-37 (2015).

[12] Q.K. Wan and H.K. Lo, "A Mixed Integer Formulation for Multiple-Route Transit Network Design", *Journal of Mathematical Modelling and Algorithms* 2(4), 299-308 (2003).

[13] A.G. Marin, and P. Jaramillo, "Urban Rapid Transit Network Design: Accelerated Benders Decomposition", *Annals of Operations Research* 169(1), 35-53 (2009).

[14] A. Barra et al, "Solving the Transit Network Design Problem with Constraint Programming", 11th World Conference in Transport Research, Berkeley, June 2007.

[15] R. Borndorfer, M. Grotschel, and M.E. Pfetsch, "A Column-Generation Approach to Line Planning in Public Transport", *Transportation Science* 41(1), 123-132 (2007)

[16] D. Feillet et al, "A Note on Branch-and-Cut-and-Price", *Operations Research Letters* 38(5), 346-353 (2010).

A Passenger-Centric Approach to Air Traffic Flow Management

Alexandre Jacquillat
Heinz College, Carnegie Mellon University
Email: ajacquil@andrew.cmu.edu

1 Introduction

Transportation networks often involve two distinct layers: vehicles vs. end users. For instance, public transit systems operate subways and buses to transport riders; logistic systems operate container ships, cargo aircraft and delivery trucks to transport packages; and air transportation systems operate aircraft to transport passengers. Although interconnected, these layers do not always coincide due to multi-leg itineraries involving connections between vehicles. Extensive routing and flow management research has focused primarily on the optimization of vehicle operations. However, this might not result in optimal outcomes from end users' perspectives when travel itineraries involve connections between multiple vehicles (e.g., multi-line itineraries in public transit, multi-modal deliveries in logistics, and multi-leg passenger itineraries in air transportation).

We develop an original user-centric approach to traffic flow management, with a focus on Air Traffic Flow Management (ATFM). ATFM consists of controlling the flows of aircraft across air traffic operations networks to mitigate congestion costs. Specifically, it optimizes flight operations at capacitated airports and through capacitated air traffic control sectors to absorb delays at departure airports or in the en-route airspace rather than in the terminal airspace at the arrival airport, where they are most costly to operate from safety, economic and environmental perspectives. Successful implementation of ATFM initiatives in practice has enabled significant reductions in congestion costs faced by airlines, airports and passengers [Ball et al., 2007, Vossen et al., 2012].

Existing ATFM developments are based on flight delay considerations exclusively. However, the costs of congestion do not depend only on the magnitude of flight delays, but also on their impact on passenger itineraries. First, the same levels of flight delays can induce higher passenger costs if they are borne by flights carrying more passengers. Second, flight delays can create disproportionate disruptions for passengers traveling on multi-leg itineraries if they result in misconnections. In fact, passenger delays increase non-linearly with flight delays, and this effect is amplified by such

factors as congestion at connecting airports, high load factors, and limited flight frequency in some markets [Barnhart et al., 2014]. From a system-wide standpoint, nearly 50% of congestion costs are borne by passengers, mostly driven by 2% to 5% of itineraries being disrupted due to flight cancellations or missed connections [Ball et al., 2010]. Therefore, the consideration of passenger itineraries can significantly impact the ATFM outcomes and resulting costs of air traffic congestion.

We propose a joint analytical and computational approach to balance the costs of vehicle delays (e.g., flight delays) and user delays (e.g., passenger delays) in traffic flow management. First, an analytical Markov Decision Process model derives structural insights on the drivers of user-centric operations. Second, a large-scale integer programming optimizes ATFM operations in large-scale traffic networks, while tracking their impact on passenger accommodations and delays. An original rolling procedure decomposes the problem over time while ensuring global feasibility. It is shown to enable the model's implementation in short computational times. Computational results in the US National Aviation System suggest that large reductions in passenger delays can be achieved at comparatively small increases in flight delay costs. Analytical and computational results highlight two main levers of user-centric operations: (i) *delay allocation*, which determines *which* flights to delay or prioritize to minimize passenger delays, and (ii) *delay introduction*, which deliberate adds departure holds to avoid passenger misconnections.

2 Analytical Model of User-centric Operations

We consider a facility with a set of arriving vehicles and a set of departing vehicles. Users fall into three categories: (i) departing users (i.e., users traveling in a departing vehicle), (ii) arriving users (i.e., users traveling in an arriving vehicle), and (iii) connecting users (i.e., users transferring from an arriving vehicle to a departing vehicle). Any time an incoming vehicle arrives at the facility, all connecting users whose second-leg vehicle has already left are re-accommodated on the next available vehicle serving the same destination. The decision-making problem determines, at any point in time, which departing vehicle to operate at the facility, if any.

The problem is formulated as a continuous-time Markov decision process. The model's parameters capture the schedule of arriving vehicles, the frequency of re-accommodation options on each origin-destination market, and the departure capacity of the facility. Decisions are made as a function of a state variable that captures the sets of incoming vehicles that have arrived already and the set of departing vehicles that have already left the facility. The formulation minimizes total discounted user delay, including the delay borne by users waiting for a vehicle to depart, and the added travel time borne by misconnecting users.

The characterization of the optimal policy outlines the core trade-off in user-centric operations, between minimizing wait times by operating departing vehicles as soon as possible, on the one hand,

and avoiding misconnections by holding departing vehicles at the facility, on the other. This takes place through two main mechanisms: (i) *delay allocation* (i.e., prioritizing some vehicles among the set of departing vehicles), and *delay re-allocation* (i.e., updating departing vehicle priorities upon any vehicle arrival), and (ii) *delay introduction* (i.e., deliberately holding departing vehicles to avoid user misconnections). Results also identify the main drivers of the decisions regarding *which* vehicle to prioritize and whether to operate *any* vehicle at all. First, it is more beneficial to operate departing vehicles with more users ready to depart but fewer incoming connections. Moreover, the faster incoming vehicle arrivals are expected, and the more frequent re-accommodation opportunities, the stronger the incentives to operate any departing vehicle.

3 Integer Programming Model of Passenger-centric ATFM

We then augment ATFM optimization models by explicitly accounting for the impact of flight operations on passenger accommodations across air traffic networks.

The model takes as inputs: (i) the schedule of flights across the network of airports, (ii) aircraft itineraries, (iii) the operating capacity of each airport, and (iv) passenger itineraries. It optimizes flight operations (i.e., departure and arrival times) in a capacitated network of airports, while tracking resulting passenger accommodations and passenger delays. In particular, the formulation identifies disrupted itineraries, and re-allocates passengers to later flights whenever a connection is missed based on re-accommodation options and aircraft capacities. The objective function comprises aircraft delay costs and passenger delay, with a weight parameter ρ that trades off the two objectives. Specifically, the model is formulated as follows:

$$\min \quad \text{Flight delay costs } + \rho \cdot \text{ Total passenger delay} \tag{1}$$

$$\text{s.t.} \quad \text{Flight operating constraints} \tag{2}$$

$$\text{Airport capacity constraints} \tag{3}$$

$$\text{Passenger flow constraints} \tag{4}$$

We develop a rolling algorithm to solve the model. In other words, we optimize traffic flows for a restricted look-ahead window (set to 4 to 6 hours) iteratively over time (every hour). This approach is consistent with current practice and with the recent literature [Bertsimas et al., 2011]. The passenger-centric considerations, however, complicate the design of this rolling procedure because of the need to capture the impact of flight operations on passenger itineraries across the full day. We therefore propose additional constraints to maintain global feasibility and to ensure consistency of passenger flows from one time period to another.

We implement the model in the US National Aviation System using real-world data on flight schedules, passenger itineraries and airport capacities. We create test instances with up to 30

airports subject to ATFM interventions, which captures the largest instances encountered in practice. At each iteration of the rolling algorithm, the model optimizes the operations of up to 14,000 flights and the accommodation of passengers booked on up to 60,000 itineraries. Extensive computational experiments show that the model can be solved with a median runtime of 1-3 minutes at each iteration—a moderate increase from baseline models where passenger flows are omitted. This computational performance enable the implementation of passenger-centric approaches in practice.

Results suggest that significant ATFM improvements can be achieved by incorporating passenger considerations into flight optimization algorithms. Indeed, the passenger-centric approach to ATFM developed here permits large reductions in passenger delays at comparatively small increases in flight delay costs, as compared to baseline models that do not consider passenger delays. These are primarily driven by a sharp reduction in passenger misconnections and, to a lesser extent, by the reduction of delays borne by non-stop passengers. These improvements are obtained by prioritizing flights carrying more non-stop passengers, flights with more outgoing connections (i.e., flights booked by more passengers as the first leg of connecting itineraries) and flights with fewer incoming connections (i.e., flights booked by fewer passengers as the second leg of connecting itineraries). Vice versa, the other flights are de-prioritized, or even held deliberately on the ground if the corresponding benefits of avoiding misconnections outweigh the associated delay increases. These results confirm the structural insights from our analytical model in large-scale networks.

These results suggest that enhancing ATFM initiatives by explicitly accounting for passenger itineraries could provide significant benefits to airlines and passengers. The success of Collaborative Decision Making provides a framework to facilitate the sharing of passenger-level information and its integration into decision support systems to make ATFM more beneficial to all stakeholders.

References

[Ball et al., 2010] Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., and Zou, B. (2010). Total Delay Impact Study. Technical report, National Center of Excellence for Aviation Operations Research, College Park, MD.

[Ball et al., 2007] Ball, M., Barnhart, C., Nemhauser, G., and Odoni, A. (2007). Air Transportation: Irregular Operations and Control. In Barnhart, C. and Laporte, G., editors, *Handbook in Operations Research & Management Science*, volume 14, pages 1–67. Elsevier.

[Barnhart et al., 2014] Barnhart, C., Fearing, D., and Vaze, V. (2014). Modeling Passenger Travel and Delays in the National Air Transportation System. *Operations Research*, 62(3):580–601.

[Bertsimas et al., 2011] Bertsimas, D., Lulli, G., and Odoni, A. (2011). An Integer Optimization Approach to Large-Scale Air Traffic Flow Management. *Operations Research*, 59(1):211–227.

[Vossen et al., 2012] Vossen, T., Hoffman, R., and Mukherjee, A. (2012). Air Traffic Flow Management. In *Quantitative Problem Solving Methods in the Airline Industry*, volume 169 of *International Series in Operations Research & Management Science*, pages 385–453. Springer US.

General solution scheme for the Static Link Transmission Model

Mark P.H. Raadsen
Institute of Transport and Logistics Studies
University of Sydney

Michiel C.J. Bliemer
Institute of Transport and Logistics Studies
University of Sydney

Mark P.H. Raadsen (corresponding)
Institute of Transport and Logistics Studies
University of Sydney
Email: mark.raadsen@sydney.edu.au

1 Introduction

To date, most static traffic assignment models remain neither capacity constrained nor storage constrained, i.e. they let flow exceed the link capacity and do not consider spillback. In this work we consider a recently proposed static assignment model formulation that is both capacity as well as storage constrained [1]. The formulation of this model is derived – and consistent with - a state-of-the-art continuous time, first order, dynamic network loading model proposed in [2]. The importance of being able to capture spillback effects in a rigorous way in static assignment cannot be overstated. Not doing so typically results in underestimation of path travel times. This is especially true regarding paths that are affected by queues that spill back, even though these paths might not traverse the bottleneck(s) that caused the intial formation of the queues. As far as the authors are aware, the only other static models that attempt to capture spillback are found in [3] and [4]. However, in their work, they assume stable queues that are not the result of the adopted (steady-state) flow rates, necessitating the assumption that the queue originated from an - unmodelled - preceding period, which is not ideal, although mathematically convenient. In our work, the queues are consistent with the steady-state flow rates obviating the need to "choose" the queue, instead they follow from the link inflow and outflow rates.

In our companion paper [1], the mathematical problem formulation of this novel static assignment model is described. However, similar to [3] and [4], no solution algorithm for general transport networks exists. In this paper, we do propose a general solution scheme for this model, capable of solving large-scale - real-world - networks despite the challenges introduced when incorporating path flow interdependencies due to spillback effects in a static context. As far as the authors are aware, this is a first in static assignment.

2 Methodology and model formulation

Let us consider a transport network $\mathcal{G}(\mathcal{N},\mathcal{A})$, with links $a \in \mathcal{A}$ and nodes $n \in \mathcal{N}$. The network loading consists of a link model and a node model. Unlike most static network loading models, the link model is consistent with any two-regime concave fundamental diagram (FD). The FD has an uncongested branch (I) where density increases with increasing flow, and a congested branch (II) where density increases with decreasing flows. Density [veh/km] is denoted via inverse flux functions $\Phi_{I,a}^{-1}(q_a), \Phi_{II,a}^{-1}(q_a)$, for the uncongested and congested branch respectively, with the flow rate [veh/h] denoted by $q_a \in [0, q_a^{\max}]$, see Figure 1.

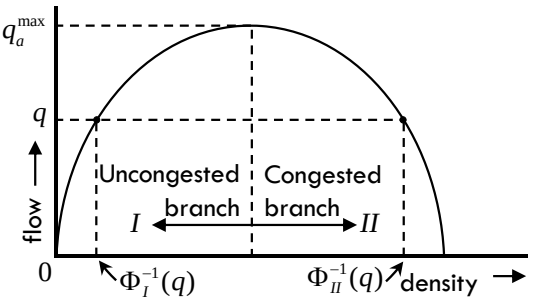


Figure 1: General concave two-regime FD consistent with static network loading model with capacity and storage constraints.

Similar to dynamic models and different to most static models, we differentiate between link inflow rate u_a and link outflow rate v_a. When $u_a < v_a$, a queue forms explicitly during simulation period T [h]. Consequently, the portion α_a of accepted flow is given by $\frac{v_a}{u_a}$. In traditional static models, $\alpha_a = 1, \forall a \in \mathcal{A}$ and $u_a = v_a$. Here, this is no longer the case.

Further, we only focus on network loading, we assume that for a given path set $p \in \mathcal{P}$, desired path flows f_p [veh/h] are known and given. Path incidence indicator δ_{ap} yields zero if link a is not on path p and one otherwise. Link set $\mathcal{A}_{ap}$ contains all links preceding link a on path p. The link model is then formulated as follows:

$$u_{ap} = \delta_{ap} f_p \prod_{a' \in \mathcal{A}_{ap}} \alpha_a, \quad \text{with } \alpha_a = \tfrac{v_a}{u_a}, \quad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}, \tag{1}$$

$$u_a = \sum\nolimits_{p \in \mathcal{P}} u_{ap}, \quad \forall a \in \mathcal{A}, \tag{2}$$

The construction of α depends on the node model function, denoted $\Gamma_n(\cdot)$, which distributes competing *sending flows* s_a [veh/h], from incoming links $a \in \mathcal{A}_n^-$, based on the available *receiving flows* r_a [veh/h] on outgoing links $a \in \mathcal{A}_n^+$, for node n. *Splitting rates* φ_{ab} convert the link sending flows to turn sending flows. Hence, the implicit (general) node model function input and outputs are defined as follows:

$$\varphi_{ab} = \tfrac{1}{u_a} \sum\nolimits_{p \in \mathcal{P}} \delta_{bp} u_{ap} \quad \forall a \in \mathcal{A}_n^-, \forall b \in \mathcal{A}_n^+, \forall n \in \mathcal{N}, \tag{3}$$

$$(\mathbf{u}_n, \mathbf{v}_n) = \Gamma_n(\mathbf{s}_n, \mathbf{r}_n, \boldsymbol{\varphi}_n), \quad \mathbf{u}_n = [u_b]_{b \in \mathcal{A}_n^+}, \mathbf{v}_n = [v_a]_{a \in \mathcal{A}_n^-}, \mathbf{s}_n = [s_a]_{a \in \mathcal{A}_n^-}, \mathbf{r}_n = [r_b]_{b \in \mathcal{A}_n^+}, \boldsymbol{\varphi}_n = [\varphi_{ab}]_{a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+}, \tag{4}$$

The link and node model interact by transforming link inflow rates to (downstream) sending flows - restricted by the link's capacity – and constructing the receiving flows which are conditional on the link outflow rates, the available storage capacity, and link capacity via:

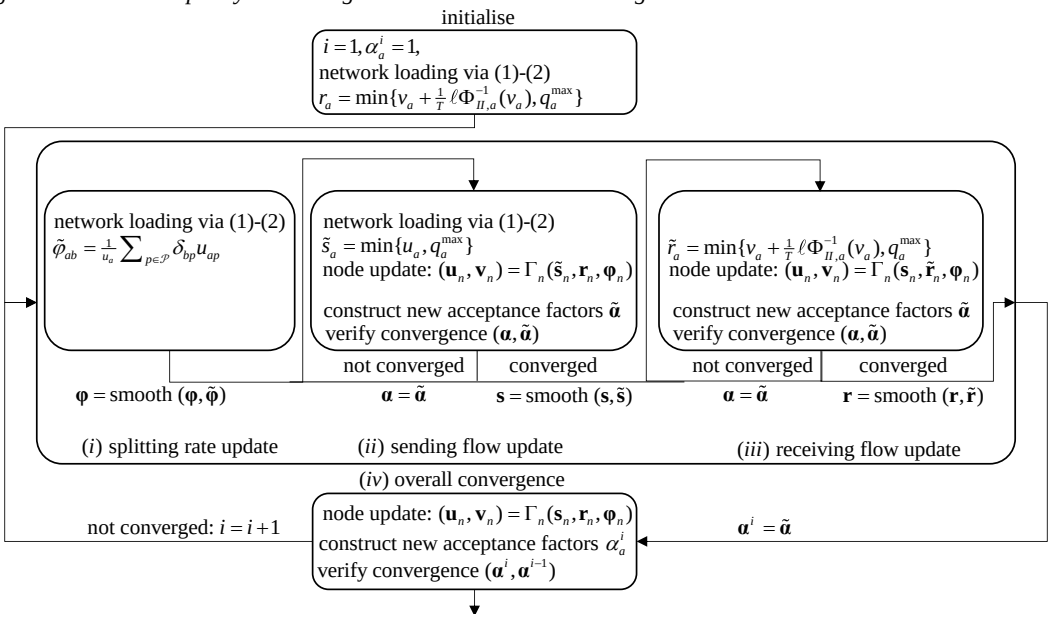
$$s_a = \min\{u_a, q_a^{\max}\}, \quad \forall a \in \mathcal{A}, \tag{5}$$
$$r_a = \min\{v_a + \tfrac{1}{T}\ell\Phi_{II,a}^{-1}(v_a), q_a^{\max}\}, \quad \forall a \in \mathcal{A}, \tag{6}$$

with link length ℓ [km] and simulation duration T scaling the - outflow based - density to the appropriate storage capacity. The storage capacity supplemented with the outflow rate dictates the possible receiving flow, see [1] for the original derivation of this model. In this work, for the first time, we go beyond the mere formulation and provide a general solution scheme, which is discussed conceptually in the remainder of this extended abstract.

3 Solution scheme

While we know a solution exists to this model formulation (see [1]), finding such a solution is far from trivial given that the inflows and outflows depend on the sending and receiving flows via the node model. At the same time, the sending and receiving flows depend on the inflows and outflows creating a circular dependence in which all link flows can depend on all other links in the network. In a simplified case, when omitting storage constraints, i.e. fixing $r_a = q_a^{\max}$, the point queue model of [5] results. In this earlier work, solutions are constructed via a relatively straightforward fixed point algorithm that in most cases converges quickly but may struggle to converge in some rare cases where multiple solutions exist. Adding storage constraints significantly complicates matters since the receiving flow is no longer constant, leading to an increased instability of the algorithm. We address these issues by proposing a solution scheme - see Algorithm 1 - that revolves around the three inputs of the node model, namely (i) splitting rates, (ii) sending flows, (iii) and receiving flows.

Algorithm 1: Static capacity and storage constrained network loading solution scheme.



Solutions to each of the three components is found separately, while temporarily fixing the other two components. This maximises algorithm stability while searching for a solution. Also, we can demonstrate that for each of the three sub-problems a solution exists. We then iteratively solve each sub-problem until convergence between sub-problems is reached as well. This then constitutes the solution to the overall network loading problem. The sending flow and receiving flow sub-problems

are themselves fixed point algorithms that require an iterative scheme to solve. In [5], steps (i) and (ii) are combined while step (iii) does not exist. By separating out steps (i) and (ii) we demonstrate that we can solve problems with multiple solutions that up untill now did not converge, an example of which is originally described in [5]. Further, the introduction of step (iii) ensures storage constraints and spillback effects are properly captured in a static context. In the full paper we demonstrate that smoothing between the various sub-problems is still required in general networks to ensure overall convergence, hence the verification on the smoothed results via step (iv).

4 Case studies

We investigated a number of case studies on both hypothetical and real-world networks to illustrate the effects and potential benefits of modelling spillback in a static context, conduct parameter calibration and investigate/compare computation costs. To illustrate the differences between traditional assignment, a point queue model and our newly proposed model, consider a locally oversaturated grid network with uni-directional links as depicted in Figure 2 (fixed demand, AON assignment). The bottleneck links found in traditional static assignment are depicted in Figure 2(a).

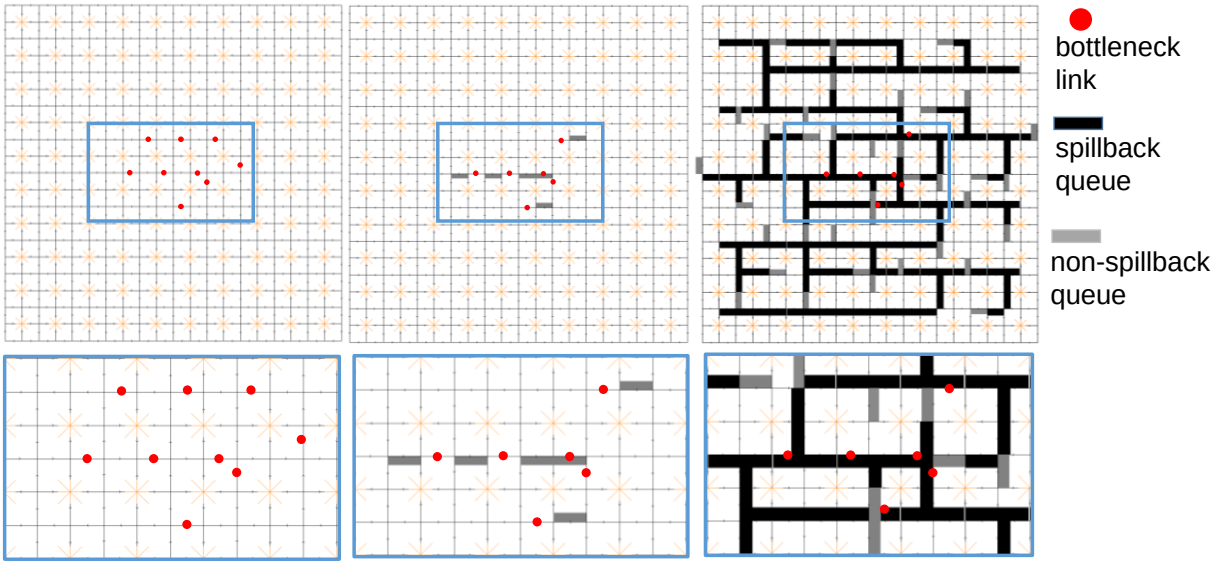


Figure 2: (a) Traditional static assignment bottleneck links, (b) point queue results as per [5], no spillback. (c) Static LTM with storage constraints causing spillback..

Note that this does not yield explicit queues, it only results in flows exceeding capacity on such links. Figure 2(b) depicts the point queue model results as per [5], where queues emerge in front of bottleneck links. The Point queue model explicitly withholds excess flow resulting in less bottleneck links compared to traditional static assignment, however it does not consider spillback. Figure 2(c) shows the result of imposing storage constraints via our newly proposed *Static Link Transmission Model*, illustrating how queues spill back across the network. This is especially noticeable in highly saturated conditions as is the case in this example. Finally, we point out that queues can potentially spill back all the way to their original bottleneck link. This – when needed – is also be captured by our approach. In that case circular dependencies cause severe deterioration of the network performance and are especially difficult to solve. Details on the algorithm, extensions and further results are discussed in the full paper.

References

[1] Bliemer, M.C.J., Raadsen, M.P.H., 2018 (*submitted to ISTTT 23*). Static traffic assignment with residual queues and spillback: model formulation. ISTTT 23, 24-26 July 2019, Lausanne Switzerland.

[2] Bliemer, M.C.J., Raadsen, M.P.H., *in press*. Continuous-time general link transmission model with simplified fanning, part I: Theory and link model formulation. Transp. Res. Part B Methodol. 0, 1–29. doi:10.1016/j.trb.2018.01.001

[3] Smith, M., Huang, W., Viti, F., 2013. Equilibrium in Capacitated Network Models with Queueing Delays, Queue-storage, Blocking Back and Control. Procedia - Soc. Behav. Sci. 80, 860–879. doi:10.1016/j.sbspro.2013.05.047.

[4] Smith, M.J., 2013. A link-based elastic demand equilibrium model with capacity constraints and queueing delays. Transp. Res. Part C Emerg. Technol. 29, 131–147. doi:10.1016/j.trc.2012.04.01.1

[5] Bliemer, M.C.J., Raadsen, M.P.H., Smits, E.-S., Zhou, B., Bell, M.G.H., 2014. Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. Transp. Res. Part B Methodol. 68, 363–384. doi:10.1016/j.trb.2014.07.001.

Dynamic Flexible Time Window Pricing for Attended Home Deliveries

Charlotte Köhler

Jan Fabian Ehmke

Management Science Group

Otto von Guericke University, Magdeburg, Germany

Email: charlotte.koehler@ovgu.de

Ann Melissa Campbell

Department of Management Sciences

University of Iowa, Iowa City, United States

Catherine Cleophas

Service Analytics Group

Christian Albrechts University, Kiel, Germany

1 Introduction

Many products that are ordered online, such as groceries, require the customer to be present during the delivery to home or office. For these attended home deliveries (AHD), retailers offer a selection of time windows on their website, and customers choose one of these windows during the order process. A significant challenge when creating time window offer sets lies in presenting suitable time window options for customers. First, customers want convenient short time windows, but short time windows can significantly restrict the ability to accept future requests and decrease the flexibility of the route plan [4]. Second, demand for time windows is usually highly imbalanced and leads to scarce delivery capacities, especially in after work hours. Third, customers want to pay as little as possible for their delivery, and profit margins are low. Major online supermarkets like *Tesco* (United Kingdom) or *Bringmeister* (Germany) offer both long and short time windows and differentiate these options through different delivery fees. Low prices are used to nudge customers towards accepting longer or less popular time windows. However, online supermarkets usually set fees that are static over the whole booking process instead of dynamically updating them to

already accepted orders and routing characteristics.

In this work, we introduce the idea of dynamic flexible time window pricing. We consider the impact of offering time windows of different lengths on the route plan's *flexibility* and *dynamically* adjust the fees per time window over the booking process. The literature offers various approaches for pricing time windows of uniform length, e.g., through anticipating expected delivery costs [5]. However, these approaches neither consider the current flexibility of a route plan nor the idea of including time windows of multiple lengths in the same offer set. In [2], we investigated the idea of flexible time window management and developed simple ways to preserve flexibility of the route plan by strategically offering short time windows. However, we did not set prices for time windows of different lengths.

In this work, we want to develop and analyze rules that optimize delivery prices to maximize the fee revenue for the retailer. We consider two time window lengths, long and short, and these time windows are priced differently according to their length and their impact on route flexibility. From a routing perspective, the largest number of deliveries is possible with long time windows, but customers prefer the better service of the short windows and do not always accept a long time window offering, even when it is cheaper. We develop different flexible pricing schemes, which are easy to adapt by an online retailer. To analyze their effectiveness, based on recent literature, we incorporate realistic customer behavior and introduce a choice model for time windows of multiple lengths. We investigate the presented schemes with a case study for an online supermarket showing the advantages of dynamic flexible pricing compared to static time window pricing.

2 Creating Flexible Time Window Offer Sets

For each request j, we create an offer set O_j based on spatio-temporal customer attributes as well as route plan information. O_j can include both long and short time windows. We always offer all feasible time windows, but assign them different fees. We consider two sets of time windows: Set S contains short time windows with length s, and set L contains long time windows with length l. Our approach maintains a tentative route plan for each vehicle based on the already accepted requests. We use an insertion-based heuristic as presented in [1] to evaluate the feasibility of inserting a new request j within the tentative route plan. We compute *time spans* that reflect feasible ranges for start and end of service times for request j at the insertion position between customer i and $i+1$ on a vehicle. For each feasible insertion position, we create sets that contain all feasible time windows S' and L' and merge them into a single offer set O_j.

While long time windows are less likely to be accepted by a customer but provide greater flexibility for accommodating future requests, we will offer the long time window options always for free. To determine a request-dependent delivery price for the short time windows, we adapt

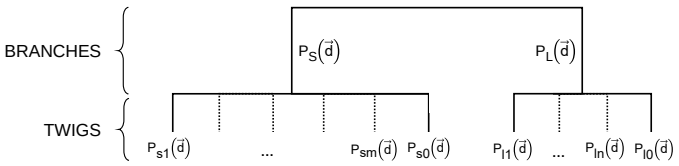


Figure 1: Nested Logit Model for Customer Choice of Long and Short Time Window

and refine flexibility mechanisms from [2] and combine these with a nested logit (NL) model. In [2], we saw that it makes sense to maintain a high level of routing flexibility especially in the beginning of the booking process. To this end, we propose a pricing scheme that offers short time windows for a higher delivery fee to the early arriving customers to incentivize them to choose a long time window. With this *utitlization-based* scheme, late arriving customers receive short time window offerings at a lower price (but may face fewer available short time window options). We quantify this by measuring the current utilization of our service capacity and compute how much of the available service time has already been consumed by the already accepted customers when creating offer sets for the available delivery vehicles.

To represent the choice behavior of customers selecting a delivery option from an offer set of long and short time windows given different delivery fees, we incorporate a NL model [3]. Compared to the well-known multinomial logit model (MNL, [5]), the nested variant can consider different utility expressions across groups of alternatives. In other words, customers compare the utility of long versus short time windows separately from comparing the utility for time windows with different delivery fees. Figure 1 presents the probabilities for customer choice of long and short time windows. Within the first level, each branch describes the customer's probability for choosing either a short ($P_S(\vec{d})$) or a long time window ($P_L(\vec{d})$). The second level considers "twigs", which model the customer's probability of choosing a specific alternative from one of the "nests" (including the no purchase options $P_{s0}(\vec{d})$ and $P_{l0}(\vec{d})$). The twig selection relies on the MNL model and its parameters as presented by [5].

For the branch selection, we need to assess the offer set in terms of its overall suitability regarding the short time window offering. We assume that – if offered at same terms – customers always prefer short over long time windows. We define the probability of choosing a short time window as shown in Formula 1, which is a function of the price of the short time window $\vec{d}$. We consider the customer's price sensitivity β_d as well as his/her sensitivity to the length of the time window β_{length}, which defines the price level a customer would prefer long time windows.

$$P_{short}(\vec{d}) = \frac{\sum_{s \in S'} exp(\beta_0 + \beta_s + \beta_{length} * \beta_d * d_s)}{\sum_{s \in S'} exp(\beta_0 + \beta_s)} \tag{1}$$

The branch probability for a long time window is then $P_{long} = 1 - P_{short}(\vec{d})$.

	10€	9€	8€	7€	6€	5€	4€	#acc	#240	#30	feeRev
Static	100%							63.6	34.7	28.9	289€
Dynamic Flexible	60%		20%		20%			62.2	29.5	32.7	302€
Dynamic Flexible	80%				20%			63.3	32.1	31.2	299€
Dynamic Flexible	40%	40%			20%			62.4	30.2	32.2	296€
Dynamic Flexible	40%	40%	20%					63.9	32.5	31.3	295€
Dynamic Flexible	80%						20%	63.0	30.9	32.1	294€

Table 1: Comparing Static and Dynamic Flexible Time Window Pricing

3 Experiments

We investigate the effectiveness of dynamic flexible time window pricing by simulating the booking process for an online supermarket in Berlin, Germany. We assume a fixed delivery capacity of three vehicles. In one set of experiments, we set the lengths of short time windows to 30 minutes and long windows to 240 minutes and offer up to 16 short and 2 long time windows. We use real travel times provided by *OpenStreetMap* and real data from [5] for customer choice calibration.

Table 1 shows an example result based on the average of 1,000 simulations for a dynamic flexibility pricing as well as a static pricing of short time windows. For the static pricing, we can see that if short time windows are offered at 10 € to all customers, we accept 63.6 customers in total with 34.7 in a long and 28.9 in a short time window, creating a fee revenue of 289 €. The remaining rows show how further variants of the dynamic pricing schemes perform. With this, we measure the utilization of the service capacity during the booking process and decrease the time window fees as the number of accepted orders grows. For example, in the second row, we offer short time windows for 10 € until 60% of the available service capacity has been assigned, then reduce the fee to 8 €, and finally offer short time windows for only 6 € until the remaining 20% capacity have been utilized. This increases the fee revenue by 5% compared to static pricing and allows the firm to accept four additional orders for short time windows while keeping the total number of accepted orders constant.

4 Outlook

In the conference presentation, we plan to discuss further variants of dynamic pricing schemes, highlighting the gains from considering the route plan's flexibility in the dynamic pricing of flexible time windows. Further pricing schemes are based on customer characteristics (location, time window preferences, basket value) as well as characteristics of the evolving route plan (e.g., available capacity). We will analyze the best schemes for setting fees and the range of fee values required to maximize revenue. We show that the proposed schemes work well and outperform current static pricing strategies of online retailers.

References

[1] A.M. Campbell and M.W.P. Savelsbergh, "Decision Support for Consumer Direct Grocery Initiatives", *Transportation Science* 39(3), 313-327 (2005).

[2] C. Köhler, J.F. Ehmke and A.M. Campbell, "Flexible Time Window Management for Attended Home Deliveries", Working Paper Series, Otto von Guericke University, Magdeburg, Germany, 2018.

[3] F.S. Koppelman and C.-H. Wen, "Alternative nested logit models: structure, properties and estimation", *Transportation Research Part B: Methodological* 32(5), 289-298 (1998).

[4] I. Lin and H. Mahmassani, "Can Online Grocers Deliver?: Some Logistics Considerations", *Transportation Research Record: Journal of the Transportation Research Board* 1817, 17-24 (2002).

[5] X. Yang, A.K. Strauss, S.M. Christine, R. Eglese, "Choice-Based Demand Management and Vehicle Routing in E-Fulfillment", *Transportation Science* 50(2), 363-761 (2016).

Integrated airline schedule, aircraft and passenger recovery: incorporating passenger response to disruptions

Luis Cadarso

European Institute for Aviation Training and Accreditation (EIATA), Rey Juan Carlos University, Fuenlabrada, Madrid, 28943, Spain

Email: luis.cadarso@urjc.es

Vikrant Vaze

Thayer School of Engineering, Dartmouth College

14 Engineering Drive, Hanover NH 03755, United States of America

1 Introduction

During daily operations of an airline network, various incidents may cause deviations from the planned operations, sometimes making it impossible to operate the schedule as originally planned. In such situations airlines need to adjust the schedule for the time period of the incident, and then carry out further recovery steps in order to get back to original schedule. Disruptions usually require the airline to reschedule some passengers onto alternative itineraries to take them to their destinations as soon as possible. In the current airline practice, the recovery problem is often solved sequentially [1]. Once a disruption has begun and impacted an airline, one of the main decisions to make is about timetabling, i.e., whether to maintain punctuality or to delay or cancel flights. Then, the aircraft recovery problem re-assigns individual aircraft to fly the repaired schedule, while matching passengers' transportation needs with available seating on the assigned aircraft demand while satisfying the aircraft maintenance requirements. Finally, crews are rescheduled to comply with the government regulations and collective bargaining labor agreements. However, this approach has limitations [4]: computing a new timetable without accounting for aircraft and passengers considerations may produce a suboptimal timetable or even an infeasible one for aircraft assignment and passenger recovery purposes. Moreover, all prior studies ignore passengers' response to disruptions and recovered schedules [3]. Specifically, all existing literature assumes that all the rebooked passengers show up [2]. However, due to passenger compensation rules established in some regions of the world, this may not be true. Passengers' response may be significantly influenced by the operator's recovery actions and the provided level of service.

In a recovery context, level of service may be measured with the amount of delay that passengers suffer when arriving at their destination as compared to their planned arrival time. Existing regulations in some countries, which impose monetary compensations to passengers in case of disruptions, alter the way they perceive the utility of other alternatives, once the disruption has started, and also add new types of costs to airlines. These regulations require compensation which is dependent on distance flown, extent of delays, cancellations, or denied boarding. The monetary compensation

may enable passengers to use it to book another flight on another airline to reach their destination sooner. We define these passengers as phantom passengers. They are the passengers with a confirmed reservation, who decide not to show up for their recovered flight schedule in case of a particular disruption. Airlines may lose additional passengers if the recovery itineraries are not acceptable to the passengers in terms of their level of service. The inability to account for this important phenomenon may produce inefficient schedules, for example, by providing more seats than needed on certain itineraries. These issues were reported and identified as significant operational concerns by our airline partners who include major hub-and-spoke airlines in the world. Ours is the first study to explicitly account for these effects.

This study presents an integrated model for airline recovery which features schedule, aircraft and passenger recovery while, for the first time, explicitly accounting for passenger response to disruptions as a driver of the costs to an airline.

2 Mathematical model

The aim of the Integrated Model for Airline Operations Recovery (IMAOR) is to determine the optimal set of flights, aircraft and passenger recovery actions that will minimize the total cost which is the sum of fuel costs, other operating costs, flight delay costs, passenger re-accommodation costs, and passenger compensation costs. We will use a flight-based model that captures aircraft maintenance constraints in a novel way using delayed constraints generation. The model takes, as known inputs the following: airports and all feasible nonstop flight segments, slot availability, aircraft resource availability, disruption information (i.e., nature, place, time, duration, etc.), original scheduled passenger flows (by passenger type and itinerary), and passenger compensation rules. Depending on the disruption, passenger type, and compensation rules, we estimate the number of phantom passengers using a binomial logit model.

The main decision variables are the following. $x_{t,f}$ is 1 if flight f is assigned to tail t, 0 otherwise. $\gamma_f \geq 0$ is the delay absorption (in minutes) due to increased cruise speed. $\delta_f^d(\delta_f^a) \geq 0$ is the departure (arrival) delay of flight f in minutes. $h_p^{p',v} \geq 0$ is the number of passengers re-assigned from itinerary p to itinerary p' for passenger type v. z_f is 1 if flight f is canceled, 0 otherwise. $\alpha_p^{p',\zeta}$ is 1 if the arrival delay of itinerary p with respect to the planned arrival time of itinerary p' is at delay-level ζ, 0 otherwise. $\chi_{f,f'}^t$ is 1 if flights f and f' are assigned to tail t consecutively and there is maintenance opportunity which is feasible in time and space.

The objective function in (1) minimizes the sum of operating costs, extra fuel consumption cost due to increased cruise speed, flight delay cost (crew cost, maintenance cost and fuel cost), passenger re-accommodation costs (e.g., meal and hotel) and passenger delay related costs (which includes passenger compensation costs and the loss of passenger goodwill). In addition, the last term speeds up the solution process and penalizes deviations from the undisrupted schedule given by $\hat{x}_{t,f}$. Note that the last two terms in the objective function are non-linear.

$$\min z = \sum_{t\in T}\sum_{f\in F_t} oc_{t,f}\, x_{t,f} + \sum_{f\in F} fc_f \gamma_f + \sum_{f\in F} dc_f \delta_f^a + \sum_{\upsilon\in\Upsilon}\sum_{p\in P}\sum_{p'\in CO_p} rc_p^{p'} h_p^{p',\upsilon}$$
$$+\sum_{p\in P}\sum_{p'\in CO_p}\sum_{\upsilon\in\Upsilon}\sum_{\zeta\in Z} pc_p^{p',\zeta}\alpha_p^{p',\zeta} h_p^{p',\upsilon} + \kappa\sum_{t\in T}\sum_{f\in F_t}\left|x_{t,f} - \hat{x}_{t,f}\right| \quad (1)$$

This objective function is minimized subject to many constraints such as itinerary delay and compensation constraints, slot availability constraints, flow conservation constraints, fleet size constraints and maintenance constraints. Here, we show only some of them, due to the limited space.

$$\sum_{t\in T}\sum_{f\in F_t} x_{t,f} + z_f = 1 \quad \forall f \in F \quad (2)$$

$$bh_t \geq \sum_{f\in F_t\cap F_\pi} h_f x_{t,f} - \sum_{f,f'\in F_t}\sum_{(f,f')\in MO}\sum_{f'\in F_\pi} rh_t \chi_{f,f'}^t \quad \forall t \in T_m, \pi \in \Pi_m \quad (3)$$

$$\sum_{p'\in CO_p} h_p^{p',\upsilon} = n_p^\upsilon \quad \forall p \in P, \upsilon \in \Upsilon \quad (4)$$

$$\sum_{t\in T} q_t x_{t,f} \geq \sum_{\upsilon\in\Upsilon}\sum_{p\in P}\sum_{p'\in CO_p : p' \ni f} (1 - \sum_{\zeta\in Z}\theta_{p,\upsilon}^{p',\zeta}\alpha_p^{p',\zeta}) h_p^{p',\upsilon} \quad \forall f \in F \quad (5)$$

Constraints (2) require that every flight must be either flown using exactly one aircraft or must be canceled. Constraints (3) ensure that tails needing maintenance do not exceed the available number of block hours. Constraints (4) are itinerary demand constraints; they determine passenger re-assignment (including that to the null itinerary). Constraints (5) are flight seating capacity constraints. The parentheses on the right-hand side of the constraints have different terms representing recapture rates for the different cases of delays: 1 if there is no delay and $1-\theta_{p,\upsilon}^{p',\zeta}$ for each level of delay ζ and passenger type υ. Note that constraints (5) feature non-linear terms in their right-hand sides.

3 Case study results

Computational experiments are based on realistic cases drawn from IBERIA's network, which features a pure hub-and-spoke network, with 48 airports and 164 OD pairs. The only hub is located in Madrid. There are five different fleet types available for these case studies: A-319, A-320, A-321, A-340-200 and A-340-600 featuring 141, 171, 200, 254 and 342 seats, respectively. A planning period of a little over three days is considered. It is divided in 4530 time instances of 1 min each. There are 1074 flights in the planning period and no flight bypasses the hub airport. We evaluate our model's performance with case studies focusing on two different disruptions: the first one features a small-scale disruption, a delay of a flight from JFK to MAD (based on data from real operations), and the second one a large-scale disruption, an airport closure of 3 hours. We set up a time limit of 600 seconds for all the computational experiments, solving all the models to at most a 2.39% optimality gap.

Table 1 shows solutions to the aforementioned small-scale disruption case study. The results for the large-scale disruption case study are omitted due to space limitations. It has four different columns. The first one is the header column. The second column displays the values as provided by the optimization model solution for the case study featuring a JFK-MAD flight delay (the small-scale disruption) assuming that each flight has the expected number of passenger no-shows. The third column lists the values given by a Monte Carlo simulation for the same optimization model solution which models each passenger agent's behavior using a stochastic simulation. The fourth column lists

the corresponding values obtained for airline actual operational decisions during the delay disruption. The optimization model is based on the expected no-show rate as an input to the passenger recovery problem, and hence is not able to fully capture all the passenger demand dynamics. Consequently, a passenger simulation is performed once the schedule for airline operations has been determined with the presented optimization model.

Table 1: Optimization and simulation solutions

	Small-Scale Disruption Case Study		
Item \ Case study	**Optimization**	**Simulation**	**Real**
Operating costs	1.329950E+7	-	1.352463E+7
Extra Fuel costs	18523.21	-	6248.144
Delay costs	25727.0	-	35217.357
Re-accommodation costs	14100	15800	17800
Compensation costs	261160	262827	281486
Rejection costs	0	0	0
# of changes	13	-	8
Expected # Phantom Pax	14.15	18.72	48.39

4 Conclusions

This study addresses the challenge of optimizing an airline's schedule, aircraft and passenger recovery in an integrated manner while explicitly accounting for cruise speed changes and previously unaccounted aspects of passenger compensations and no-shows. We develop a flight-based non-linear mixed-integer programming model using novel maintenance constraints and solve it in small computational time to reasonable optimality gaps. Our research enables incorporating thus far uncaptured but critical passenger dynamics to help airlines recover their operations more effectively and efficiently.

References

[1] C. Barnhart and V. Vaze. Irregular operations: Schedule recovery and robustness. In P. Belobaba, C. Barnhart and A. Odoni, editors, The Global Airline Industry, chapter 10. John Wiley & Sons, Hoboken, N.J., second edition, 2015.

[2] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. Journal of Scheduling, 9(3):279-298, 2006.

[3] L. Cadarso, Á. Marín, and G. Maróti. Recovery of disruptions in rapid transit networks. Transportation Research Part E: Logistics and Transportation Review, 53:15-33, 2013.

[4] J.D. Petersen, G. Sölveling, J.P. Clarke, E.L. Johnson, and S. Shebalov. An optimization approach to airline integrated recovery. Transportation Science, 46(4):482-500, 2012.

STABLE PRIMAL NUMERICAL METHOD FOR THE BOTTLENECK MODEL

Hillel Bar-Gera

Dept. of Industrial Engineering and Management

Ben-Gurion University of the Negev

Email: bargera@bgu.ac.il

1 Background

Departure-time choice is one of the most critical factors affecting congestion in transportation networks. Many studies on this topic rely on schedule penalties, as introduced by Vickery [1] in his seminal paper about the bottleneck model. Small [2] provides a thorough overview of the research that stemmed from the bottleneck model. Analytic closed form solutions are possible in many cases, but not always, for example if arrivals are stochastic [3]. In the latter case a dual method is used, focusing on the generalized cost as the main solution variable. Limitation of dual methods are discussed in section 2.

Some studies argue that there may be connections between the iterative process in computational methods and the day-to-day dynamics in practice. Unfortunately, Iryo [5] showed that if the bottleneck model is combined with a certain reasonable reaction mechanisms, the resulting dynamic system is unstable. Guo et al. [4] proposed bounded rationality as a remedy for the issue of stability in the bottleneck model. The potential of this approach is examined in section 3.

Computational methods examined here solely by their performance in terms of efficiently approximating the equilibrium at bottlenecks, ignoring possible connections to behaviors of travelers. Such methods can be useful for computational purposes, and for examining mathematical stability of the model, which requires (by definition) the existence of at least one converging process. A specific method and its numerical evaluation are presented in section 4.

2 Dual methods and their limitations

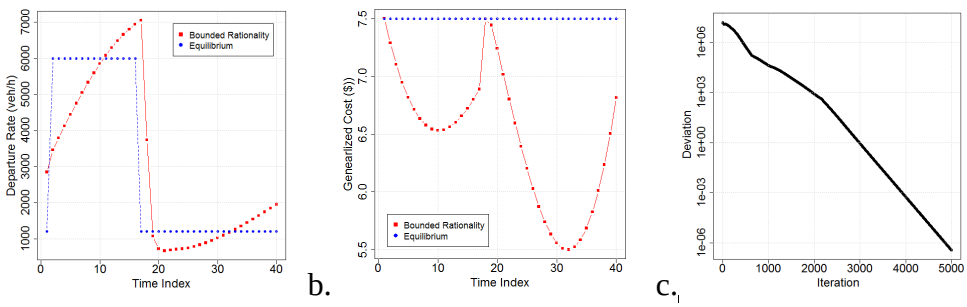
Dual methods focus on the equilibrium generalized-cost. In each iteration, a specific value is presumed for the equilibrium generalized cost. Within-day time intervals are examined sequentially. If the generalized cost at the end of the interval is higher than the presumed equilibrium value even without any departures, then the departure rate will be zero. Otherwise, a departure rate is chosen so that the generalized cost at the end of the interval will meet the presumed equilibrium value. At the end of the iteration, the total number of departures is identified and compared with the target demand. Based on this comparison adjustments are made to the presumed value of the equilibrium cost. In a single class model with homogeneous travelers, the total number of departures is a continuous and monotone

function of the presumed equilibrium cost. An iterative adjustment process can enable convergence (e.g. [3]).

Whether a dual approach can be used for multi-class models with heterogeneous travelers is not obvious. Consider the case of two classes of travelers with equal total demand, N, identical in all of their parameters except for their desired arrival time. Since their desired arrival time is different, their generalized cost at equilibrium will be different. Let $D_i(\pi_1, \pi_2)$ represent the total number of departures of class *i* as a function of the presumed generalized cost values of both classes. This function may be ill behaved, changing very slowly in certain areas while changing very rapidly in others. In particular, if the target arrival times of the two classes are almost the same, there may be a small threshold δ such that if $\pi_1>\pi_2+\delta$ then $D_1(\pi_1, \pi_2)\approx 2\cdot N$ and $D_2(\pi_1, \pi_2)=0$, but if $\pi_1<\pi_2-\delta$ then $D_1(\pi_1, \pi_2)=0$ and $D_2(\pi_1, \pi_2) \approx 2\cdot N$. Such situation of 'the winner takes it all' can complicate the adjustment process considerably, and make it unstable.

3 Bounded rationality and its implications

Guo et al. [4] proposed bounded rationality as a way to address stability in the bottleneck model. Figure 1 presents a replication of one of their scenarios, with total demand N=6000; value of time α=10 $/h; early arrival penalty of β=5 $/h; late arrival penalty of γ=15 $/h; and bounded rationality threshold of ε=2$. The figure includes within-day profiles of departure rates (1a), generalized cost (1b), and convergence (1c). The solution satisfies the conditions of bounded rationality, but it is quite different from the equilibrium solution under perfect rationality, as the threshold is relatively high, ~36% of the minimum cost. Changing the bounded-rationality threshold from ε=2$ to ε=1$ is not helpful, as the process becomes unstable even with step size that is 100-times smaller. As Iryo [5] showed, with perfect rationality, even the continuous dynamic system is not sable. Stability in this case is not only an issue of step size.



a. b. c.

Fig. 1: Departure rate and generalized cost profiles under bounded rationality – replication of a scenario from Guo et al. [4]. Within-day departure rates (a); Within-day generalized costs (b); Convergence of deviation (c)

4 A primal methods and its performance

We divide the modeling horizon into M time intervals of equal duration, ΔT. We assume that the departure rate within each time interval is constant, denoted by $r_{[t,\ t+\Delta T]}$. However, generalized cost is not constant within time intervals. In the bottleneck model, during most time intervals the generalized cost is linear. For simplicity, we shall assume that this is the case for all time intervals, ignoring the inaccuracy at the switch from earliness to lateness penalty. The discretized generalized cost at the beginning of time interval *t* is denoted by c_t.

Consider two consecutive time intervals within the congested period, and assume that both intervals remain congested under any flow shift from one of these time intervals to the other. Subsequently we will refer to such pair of intervals as "fully congested". Figure 2 illustrates several options for the generalized cost during the two intervals (there are of course many other options). Figures 2a and 2b are symmetric, with equal average generalized cost for both intervals. Thus, high-cost to low-cost shift policy imply no shift in these cases, even though equilibrium conditions are not satisfied. Alternatively, shifts can be based on the cost at the end of each interval. Figures 2c and 2d show situations with equal cost at the end of the two intervals, but again these situations do not represent equilibrium. Shifts from high-cost to low-cost intervals are therefore not sufficient, and other type of shifts, referred to as "smoothing" shifts, are needed.

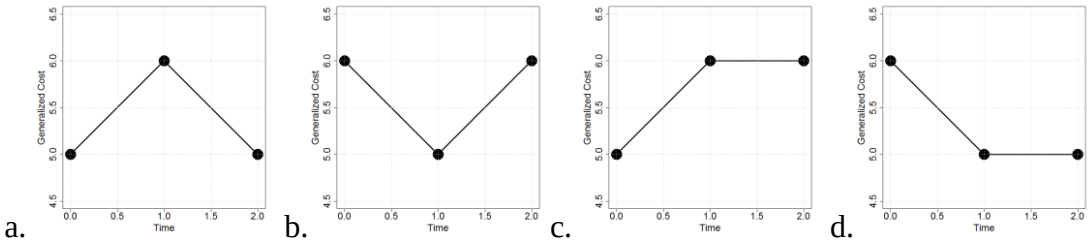


Fig. 2: Illustrative examples of generalized cost for two consecutive fully congested time intervals

Notice that under the assumption of "fully congested" intervals, a shift of flow between the two intervals will not influence the generalized cost at the beginning of the first interval, or at the end of the second interval. The only discretized generalized cost that can influenced by this shift is at the end of the first interval, which is the beginning of the second interval. The proposed smoothing shift between consecutive fully congested intervals is determined by the deviation of the generalized cost in the middle from the average of the generalized cost at the edges, that is:

$$s_t = a \cdot [\ c_t - (c_{t\ -\Delta T} + c_{t\ +\Delta T}) / 2\ ] \qquad (1)$$

where *a* is the scaling parameter. If full implementation of this shift leads to feasibility violation, the maximal possible shift is implemented. The complete method combines smoothing shifts and high-cost to low-cost shifts. Smoothing shifts are implemented if the average cost within both intervals is below the average cost of all travelers (i.e. over the entire modeling horizon). High-cost to low-cost shifts are implemented for intervals where the generalized cost at both edges is above the total average cost. Figure 3 shows the results of the proposed method in an equivalent scenario to Figure 1, with extended

modeling horizon. The figure shows that the proposed method converges to a solution that approximates the equilibrium conditions fairly well, both in terms of departure rate (3a) and in terms of generalized cost (3b). The convergence of the deviation from equilibrium (3c) is fairly slow, requiring about 200,000 iterations. Clearly, there is room for substantial improvement, but at least it shows that not all hope is lost, and that there is a chance to find primal methods that can be used to compute equilibrium solutions for the bottleneck model.

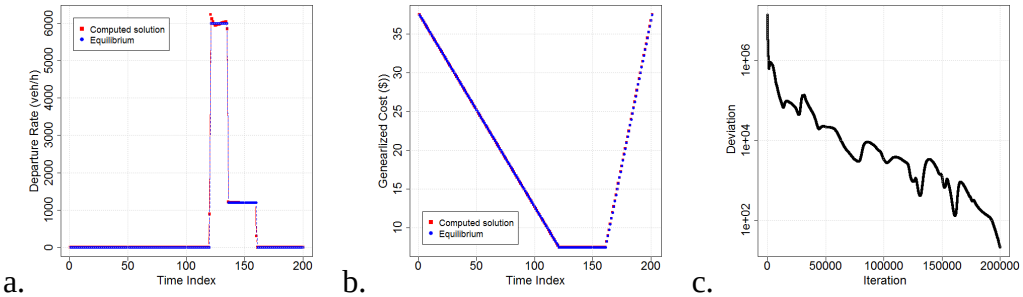


Fig. 3: Departure rate and generalized cost profiles for the proposed method.
Within-day departure rates (a); Within-day generalized costs (b); Convergence of deviation (c)

References

[1] Vickrey, W.S. Congestion theory and transport investment. *The American Economic Review* 59 (2), 251–260 (1969).

[2] Small, K.A. The bottleneck model: An assessment and interpretation. *Economics of Transportation* 4 (1–2), 110–117 (2015).

[3] van Leeuwen, D., van de Ven, P.M. Modelling user behaviour at a stochastic road traffic bottleneck. *In Proceedings of 11th EAI International Conference on Performance Evaluation Methodologies and Tools*, Venice, Italy, (VALUETOOLS 2017). https://doi.org/https://doi.org/10.1145/3150928.3150933 (2017).

[4] Guo, R.Y., Yang, H., Huang, H.J., Li, X. Day-to-day departure time choice under bounded rationality in the bottleneck model. *Transportation Research Part B: Methodological*, in press, 10.1016/j.trb.2017.08.016 (2017).

[5] Iryo, T. An analysis of instability in a departure time choice problem. *Journal of Advanced Transportation* 42 (3), 333–356 (2008).

Pricing for Drivers and Customers for Goods Deliveries

Ekaterina Alekseeva
Colisweb, France

Luce Brotcorne
INRIA , France
luce.brotcorne@inria.fr

Anton J. Kleywegt
Georgia Institute of Technology, USA.
anton@isye.gatech.edu

Youcef Magnouche
INRIA , France
magnouche.youcef@gmail.com

1 The Transportation Setting

We consider the delivery of goods, purchased through e-commerce websites or in shops, from stores or warehouses to homes in a few hours, the same day, or the next day within urban areas. The deliveries may involve in-vehicle consolidation in the form of multiple shipments on the same route, but not out-of-vehicle consolidation at transfer facilities such as crossdocks or sortation facilities. This type of service is used for various reasons, including security (e.g., expensive goods), because goods are fragile or perishable (e.g., groceries), or because goods are physically large (e.g., furniture). Such home delivery utilizes couriers who deliver with a variety of vehicles, such as bicycles, motorcycles, cars, and vans. These deliveries involve short-duration (less than a day) delivery routes, due to the characteristics of the goods (e.g., perishable goods), customer needs (e.g., urgent deliveries), and relatively small vehicle capacities.

On-Demand Delivery companies (ODDs) manage a two-sided market for the delivery of goods. They are intermediaries between consumers, retailers, and independent couriers. We consider an ODD's pricing problem in the two-sided market motivated by our collaboration with a last mile delivery company. The ODD operates a market for the delivery of goods from participating retailers to destinations specified by customers. For example, consider a customer who purchases a large item such as a piece of furniture or an appliance from a store. The customer needs the item to be delivered, and the store refers the customer to the ODD. The ODD quotes a menu of prices to the customer for delivery of the item. The prices in the menu depend on the origin, the destination, and the time window within which the customer needs the delivery to take place. The customer specifies the destination and chooses the time window for the delivery, or chooses not to accept the

delivery offer from the ODD. The customer's choice of acceptance, and the choice of time window, may depend on the prices that the ODD quotes.

On the other side of the market, the ODD contracts with independent couriers who provide the vehicles and drivers for deliveries. The ODD offers different prices in this side of the market depending on the part of the city in which and the time slot during which the courier will make deliveries. Different couriers have different preferences for the parts of the city in which to make deliveries. For example, some couriers are more flexible, possibly because they know the entire city quite well, and they are willing to make deliveries in all parts of the city, whereas other couriers have strong preferences regarding the part of the city in which they make deliveries. Different couriers also have different preferences for the time slot of the day or week in which to make deliveries. For example, some couriers have other obligations for certain times of the day or week, and they cannot or do not want to make deliveries during those times, whereas other couriers are more flexible regarding the time slot in which they make deliveries. Couriers make deliveries with different vehicle types with different capabilities — not every product can be delivered with every vehicle type. A vehicle can typically pick up multiple shipments at a store or warehouse, and deliver these shipments at their respective destinations on a route.

The compensation of each courier consists of two parts: First, the primary compensation for the time slot that the courier commits to be available to make deliveries, the part of the city in which the courier is willing to make deliveries, and the vehicle type that the courier will provide. As mentioned above, different couriers have different preferences regarding work time and part of the city in which to work, and this part of the compensation reflects these preferences relative to the customers' demands. For example, if many customers would like to receive deliveries in the evenings after work or on Saturday mornings, but few couriers want to work during these times, then the prices offered to couriers to be available during these times will be higher. Similarly, if many customers would like to receive deliveries in the core of the city, but few couriers want to make deliveries in this part of the city, then the prices offered to couriers to be available to make deliveries in the city core will be higher. Also, different vehicle types have different capabilities, and the ODD offers a higher price for vehicles with greater capabilities. Once a courier signs up to make deliveries with a particular vehicle type in a particular part of the city and a particular time slot, the courier has committed to accept all deliveries assigned to the courier in that part of the city and in that time slot, subject to the constraints of the vehicle type, and the courier is entitled to the specified compensation, whether the courier ends up being assigned any deliveries or not. The second part of the compensation of each courier is the payment for the routes that end up being assigned to the courier and the deliveries made by the courier. This secondary part of the compensation is only determined after customers have placed their delivery orders, and the ODD has assigned these delivery orders, and their associated routes, to the individual

couriers. The primary part of the compensation is intended to compensate the couriers for their delivery capability (including time) committed, whereas the secondary part of the compensation is intended to compensate the couriers for their costs incurred in making the deliveries. As mentioned before, the primary part of the compensation is driven by the couriers' work preferences relative to the customers' demands, whereas the secondary part of the compensation is determined by the estimated cost of the courier for driving a route and making deliveries. In cities with high labor cost, the secondary compensation tends to be small, about 25%, relative to the primary one.

We focus on the price planning problem in the two-sided market, that is, the problem of determining the menu of prices to quote to customers that can depend on the origin, the destination, and the time window for delivery, as well as the menu of prices to offer to couriers that can depend on the part of the city that the courier signs up to make deliveries in, the time slot that the courier commits to be available to make deliveries in, and the vehicle type that the courier will provide. These prices are selected, and the ODD and the couriers enter into their agreements, in advance of customer requests. That is, at the time that the ODD and the couriers enter into their agreements, it is not yet known exactly which deliveries will take place in each part of the city and in each time period. The ODD enters into these agreements because it needs to know that a courier will be available to make a delivery at a destination and in a time window before the ODD commits to a customer to deliver the customer's goods at that destination in that time window. Also, the courier would like to plan its own work schedule and its own compensation in advance. After the ODD has entered into agreements with various couriers, it still has an opportunity to modify the planned customer prices according to the committed delivery capacity before entering into delivery agreements with customers. For example, if the ODD failed to obtain an agreement with any courier to deliver in a particular part of the city or during a particular time window, then the ODD can exclude this part of the city or this time window from the menu of prices that it offers to customers, or it could set the price sufficiently high to pay for another delivery service.

2 Models

We use discrete choice models to model the probabilities of customers choosing particular time windows for their deliveries (or choosing not to use the ODD's delivery service). These choice probabilities depend on the menu of delivery prices offered to customers by the ODD. We also use discrete choice models to model the probabilities of couriers choosing particular parts of the city and particular time slots to commit to for making deliveries (or choosing not to commit to making deliveries). These choice probabilities depend on the menu of primary compensation prices offered to couriers by the ODD.

The detailed delivery orders are not known at the time that the ODD chooses the courier prices and customer prices. Therefore, in the pricing problem we model the aggregate delivery

capability in each part of the city and in each time window represented by the forecasted courier commitments, but we do not model the detailed vehicle routes.

We consider two price optimization problems based on multinomial logit discrete choice models. In one model, the prices are discretized, resulting in a linear optimization problem. In the other model, the prices are modeled as continuous decision variables. The basic version of this optimization problem has a nonconvex objective function. We show how to reformulate the problem as an equivalent convex optimization problem.

3 Numerical Results

Preliminary numerical results for the linear optimization problem are provided in Table 1. The instances are generated from data obtained from an ODD company. Seven categories of shipments are considered. The vehicles differ from each other by their capacity and their cost. Six vehicles are included in each type. Six time intervals with different widths are considered for couriers and for customers. In the column *Revenue* we report the difference between the planned revenue received from the customers and the planned cost paid to the couriers. Two sets of prices are considered, given by $p_i = x_i(24 - w)$ where w is the length of the time window, the idea being that if the customers offer more flexibility by choosing a larger time window for delivery, then they pay less. In each line we report the average results over 5 different instances.

The instances are solved with SCIP 6.0.0 on a computer with a processor Inter Core i5, 2.40 GHz $\times$ 4 and 8 Go of RAM. The operating system is UBUNTU 18.04.

We observe that as the number of types of vehicles increases, the solution is composed of more smaller and cheaper vehicles and thus the revenue is increasing. From the two sets of price values, given by different values of x_i, it follows that as x_i increases, the revenue decreases and the customers tend to select larger time windows. Finally, the computation time is increasing in the number of types of vehicles. Numerical instances with more than 20 types of vehicles are solved in less than 50 seconds. The results demonstrate the importance of choosing the right price levels for different time windows.

Price level	# deliveries	# vehicles	CPU Time (seconds)	Revenue	# Vehicles Sol
P1	151	2	3.68	292	10
	151	5	11.16	655.5	20
P2	151	2	3.45	185	11
	151	5	9.87	437.7	23

Table 1: Preliminary numerical results for linear optimization model.

Choice-Based Integrated Airline Fleet Assignment and Schedule Design

Chiwei Yan
Uber Technologies, Inc.
Email: chiwei@mit.edu

Cynthia Barnhart
Operations Research Center
Massachusetts Institute of Technology

Vikrant Vaze
Thayer School of Engineering
Dartmouth College

1 Introduction

Assigning aircraft types to the flight legs, a process called fleet assignment, in an airlines schedule is an important tactical decision which greatly impacts airline's profit (Barnhart et al., 2002). Such fleet assignment decisions can often be coupled with schedule design decisions by introducing a base schedule containing some mandatory and some optional flights (Lohatepanont and Barnhart, 2004). A commonly made assumption regarding customer demand, called the independent demand assumption, states that each passenger has a unique itinerary product that he/she intends to buy, and if that product is not available due to capacity constraints or revenue management policies, then that demand is simply lost. Such an assumption is not valid in practice because there always exist substitution effects among similar itinerary products. A passenger who is not able to buy his/her favorite itinerary product might choose an alternative product instead.

Motivated by this fact, in this research, we study an integrated fleet assignment and schedule design model (SD-FAM) where customer demand interactions are captured using discrete choice models (CSD-FAM). Discrete choice models are commonly used in marketing literature to model product substitutions (McFadden, 1980) and in transportation literature to model travel demand (Ben-Akiva and Lerman, 1985). They have also been widely incorporated in airline revenue management studies (Talluri and van Ryzin, 2004; Liu and van Ryzin, 2008; Gallego et al., 2014). However, there is limited research on incorporating choice models into airline planning models. Wang et al. (2014) was one of the first research studies where multinomial logit (MNL) choice model was incorporated into the fleet assignment model (FAM). As revealed in the paper, the downside of a straightforward combination of FAM with MNL choice is loss of tractability because

of the dramatic change to the structure of FAM. This issue is further exacerbated with other advanced choice models. From our own experience, for a problem instance from a major US airline, the straightforward model directly combining FAM and MNL choice does not produce even a feasible solution in 30 hours of computational time with a state-of-the-art commercial solver. This computational burden is the major obstacle preventing CSD-FAM from being applied in the airline industry. Faced by this difficulty, our research makes the following contributions to CSD-FAM:

1. We provide a tractable reformulation and reliable approximation of the choice-based integrated fleet assignment and schedule design problem that enables it to be solved for full-scale airline instances, and in doing so, achieves significant profit improvements.

2. Within this reformulation and approximation framework, we develop a novel fare-split linear program to allocate itinerary fare across corresponding flight legs and achieve significant profit improvements over commonly used heuristics, such as distance-based proration.

2 Methodology

We start with an existing reformulation called subnetwork-based FAM (SFAM) (Barnhart et al., 2009) to address CSD-FAM. The subnetwork-based FAM is an approximation scheme originally developed for efficient solution of itinerary-based FAM (Barnhart et al., 2002) with independent demand. The key idea of SFAM is to utilize composite variables to model fleet assignment decisions. In a standard FAM, binary variables $x_{l,f}$ are defined to equal 1 if fleet type f is assigned to flight l, and 0 otherwise. In SFAM, flights are first partitioned into different subnetworks. For each subnetwork, we enumerate all possible fleet assignments for all the flights in it. We then use a binary variable w_j which equals 1 if fleet assignment j is chosen for subnetwork k, and 0 otherwise. The following table shows an example of a subnetwork consisting of two flight legs (l_1 and l_2) and all possible fleet assignments with two fleet types (A and B) including the no-assignment option $\emptyset$. As can be seen from the table, there are nine possible assignments for this subnetwork, and the assignment variable w_j here indicates whether or not a particular one is selected.

Flight	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
l_1	A	A	B	B	$\emptyset$	$\emptyset$	A	B	$\emptyset$
l_2	A	B	A	B	A	B	$\emptyset$	$\emptyset$	$\emptyset$

Table 1: Illustration of the possible fleet assignment solutions for a subnetwork

With this new definition of the fleet assignment variables, our proposed reformulation represents a Dantzig-Wolfe like reformulation compared to the standard CSD-FAM formulation. The reformulation enjoys better computational efficiency because of the tightened LP relaxation bound.

On the other hand, the key challenge of SFAM is that the required number of assignment variables grows exponentially with the size of the subnetwork. Thus the size of the subnetwork determines the key trade-off between computational efficiency and solution quality, where coarser partitions and larger subnetwork sizes provide greater solution quality but higher computational requirements, and vice versa. Adding customer choice and itinerary substitution consideration leads to stronger network dependencies between flights: flights can be dependent not only because they might provide capacity for the same itinerary product, but also because the products for which they provide capacity might be jointly considered as substitutes by passengers from a specific origin-destination market. In order to extend the subnetwork-based framework to CSD-FAM and enjoy its computational advantages while still obtaining good solutions, we propose a subnetwork-based *mixed* formulation (S-CSD-FAM) where both composite variables and traditional flight-fleet assignment variables co-exist in the same model. We show that the proposed mixed formulation represents an upper bound on the original problem and develop an efficient linear program to optimize fare proration across flight legs to tighten this bound. In existing literature, such fare proration procedures are usually handled by heuristic approaches, e.g., allocating fare based on flight distances, etc. These modeling enhancements are shown to greatly improve the performance of the subnetwork-based formulation.

3 Computational Experiments

We use a full-scale daily instance from a major US airline in May 2014. The instance consists of 815 domestic flights, 4,290 itineraries leading to 47,190 total itinerary products (multiple products can correspond to the same itinerary), and 819 markets, where each market is characterized as a specific origin-destination pair. All products corresponding to the same origin and destination are assumed to be considered by the passengers in that market. So, on average, $47,190/819 \approx 57.6$ products correspond to each origin-destination market. Attractiveness value of each itinerary product and unconstrained demand for each market are estimated using airline booking and product availability data via methods described in Vulcano et al. (2012) which involve sophisticated demand untruncation and choice model estimation. The average market share of the airline under consideration across all its markets is roughly 42%. Seven different fleet types and 187 aircraft are available to operate this network. The aircraft seating capacities range from 48 to 165 passengers. In order to fully test the power of the proposed methodology, all flights are considered optional.

Table 2 illustrates results of different S-CSD-FAM runs based on different subnetwork partitions and fare split methods. It reports profit value of the solution, and CPU times if an optimal solution is found within the 12 hour CPU limit or optimality gaps at the end of solution limit. The partition is characterized using the maximum number of flights in each subnetwork (second

column) and the percentage of flights represented using the subnetwork-based formulation (third column) illustrated in Table 1. In general, we find that the solution quality (i.e., profit value) improves as the partition becomes coarser, although it is accompanied by higher computational costs. S-CSD-FAM-5 achieves the highest objective function value within the computational time limit. We also observe that the optimization-based fare proration method brings significant benefits compared to the distance-based heuristic.

Run	Partition		Fare Split	Profit	Flights	Solution
	max_subnetwork_size	composite_variable_portion		($M/day)	Selected	Time (sec)
S-CSD-FAM-0	1	100%	dist	5.160	805	9
			opt	5.167	799	9
S-CSD-FAM-1	4	100%	dist	5.413	782	63
			opt	5.450	778	40
S-CSD-FAM-2	5	100%	dist	5.517	776	205
			opt	5.517	772	226
S-CSD-FAM-3	6	100%	dist	5.578	766	795
			opt	5.618	751	1,386
S-CSD-FAM-4	4	66%	dist	5.908	732	(0.63%)
			opt	5.941	725	(0.61%)
S-CSD-FAM-5	4	33%	dist	6.119	679	(3.14%)
			opt	6.164	681	(2.94%)

Table 2: Results of different S-CSD-FAM runs (with 12 hr CPU time)

We then compare S-CSD-FAM-5, with two baseline approaches: an independent demand fleet assignment and schedule design model (ISD-FAM) and a plain implementation of CSD-FAM without the proposed reformulation and approximation scheme. Judging from the optimality gap, we can see that CSD-FAM is much more difficult to solve compared to ISD-FAM, and S-CSD-FAM-5 is more tractable than CSD-FAM, but not as tractable as ISD-FAM. In the 5 hr computational time limit, CSD-FAM is outperformed by ISD-FAM by an annual profit difference of around $15.3 million. On the other hand, S-CSD-FAM-5 significantly outperforms ISD-FAM annually by $31.4 million. This demonstrates that the tractability issues associated with CSD-FAM deteriorate its performance so much that even a simplified method (ISD-FAM), without any modeling of customer choice behaviour, outperforms it. In the 12 hr computational time limit, CSD-FAM outperforms ISD-FAM annually by $9.1 million, while S-CSD-FAM-5 further improves the annual profit by an additional $24.5 million.

Time Limit	ISD-FAM			CSD-FAM			S-CSD-FAM-5, opt		
	Profit ($M/d)	Profit Change ($M/yr)	Gap (%)	Profit ($M/d)	Profit Change ($M/yr)	Gap (%)	Profit ($M/d)	Profit Change ($M/yr)	Gap (%)
5 hr	6.059	0	0.12	6.017	(15.330)	7.74	6.145	31.390	3.52
12 hr	6.072	0	0.05	6.097	9.125	5.68	6.164	33.580	2.94

Table 3: Comparison to baseline models

References

Barnhart, C., Farahat, A., and Lohatepanont, M. (2009). Airline fleet assignment with enhanced revenue modeling. *Operations Research*, 57(1):231–244.

Barnhart, C., Kniker, T. S., and Lohatepanont, M. (2002). Itinerary-based airline fleet assignment. *Transportation Science*, 36(2):199–217.

Ben-Akiva, M. E. and Lerman, S. R. (1985). *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT Press.

Gallego, G., Ratliff, R., and Shebalov, S. (2014). A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research*, 63(1):212–232.

Liu, Q. and van Ryzin, G. (2008). On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2):288–310.

Lohatepanont, M. and Barnhart, C. (2004). Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19–32.

McFadden, D. (1980). Econometric models for probabilistic choice among products. *Journal of Business*, pages S13–S29.

Talluri, K. and van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.

Vulcano, G., van Ryzin, G., and Ratliff, R. (2012). Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, 60(2):313–334.

Wang, D., Klabjan, D., and Shebalov, S. (2014). Attractiveness-based airline network models with embedded spill and recapture. *Journal of Airline and Airport Management*, 4(1):1–25.

Dynamic speed control and lane management in the general link transmission model

Michiel C.J. Bliemer* & Mark P.H. Raadsen
Institute of Transport and Logistics Studies
The University of Sydney

Luc J.J. Wismans & Luuk J.N. Brederode
DAT.Mobility

***Corresponding Author:**
Institute of Transport and Logistics Studies, University of Sydney Business School,
The University of Sydney, NSW 2006, Australia
Email: michiel.bliemer@sydney.edu.au

1 Introduction

Dynamic network loading models for simulating traffic on networks are applied both for transport planning purposes as well as traffic management purposes. For large scale networks, macroscopic link models consistent with first order kinematic wave theory are particularly popular, especially in conjunction with first order node models. The link transmission model (LTM) is an efficient algorithm with relatively small numerical errors which is used increasingly in dynamic traffic assignment procedures. The original algorithm in [1] adopt a triangular fundamental diagram (FD). Recent extensions consider more general concave FDs ([2], [3], [4]).

For transport planning purposes, it is common to keep the FD fixed during the simulation. However, for traffic management purposes an FD may change when properties of a road segment vary over time. In particular, dynamic speed control may change the maximum speed imposed on a road segment, and dynamic lane management may open or close certain lanes. Implementation of changes in the FD in a first or order second order cell-based model can be achieved by instantaneously changing initial conditions, see e.g. [5]. While this is relatively straightforward, it assumes that all drivers on (part of the) link immediately react to the change, which may temporarily result in infeasible traffic states.[1] In LTM only boundary conditions are used and hence accounting for changes in the FD is more challenging. Variable speed limits were considered through an extension of LTM in [6] assuming a triangular FD. A more general extension to variable fundamental diagrams for LTM is presented in [7]. However, this method adds significant complexity to LTM and still suffers from possible infeasible

[1] For example, instantaneously reducing a 2-lane road to a 1-lane road may not be feasible (i.e., results in a density larger than the jam density) when vehicles are not physically able to merge onto the same lane due to existing traffic conditions.

traffic states due to the adoption of initial conditions that assume an instantaneous change of the FD across the entire link.

In this paper we therefore propose a new method that obviates the need for the assumption that the FD changes instantaneously across the entire link. Instead, we advocate an approach where the FD progressively changes in accordance with the information on the variable message sign (VMS), which travels with the speed of drivers. In other words, in case the speed limit changes from 120 km/h to 90 km/h then this will only affect drivers upstream the VMS while drivers downstream the VMS are not yet informed of the reduction in maximum speed. This effectively results in a situation where multiple FDs are active across a single link akin to multiclass traffic. The result is a behaviourly justifiable method revolving around the way information propagates (resulting in time-varying FDs) alongside the regular propagation of traffic flow. These two propagation mechanisms fit nicely into the recently proposed event-based formulations of LTM ([8] and [9]) which allow for such a separation.

2 Constrained fundamental diagram

Let the following physical parameters be given for each link: length L [km], capacity $q^{\max}$ [veh/h], jam density k^{jam} [veh/km], critical density k^{crit} [veh/km], maximum wave speed $\gamma^{\max}$ [km/h], and minimum wave speed $\gamma^{\min}$ [km/h]. For each link we assume that we consider the following fundamental relationship between flow q [veh/h] and density k [veh/km],

$$q = \Phi(k), \tag{1}$$

where $\Phi : [0, k^{\max}] \to [0, q^{\max}]$ is a continuous concave function with $\Phi(k^{\text{crit}}) = q^{\max}$, $\Phi(0) = \Phi(k^{\max}) = 0$, $q^{\max} = \Phi(k^{\text{crit}})$, $d\Phi(0)/dk = \gamma^{\max}$, and $d\Phi(k^{\max})/dk = \gamma^{\min}$. Let $\Phi_{\text{I}}(k)$ and $\Phi_{\text{II}}(k)$ denote the hypocritical branch (where flows are increasing with density) and hypercritical branch (where flows are decreasing with density) of the FD, i.e.

$$\Phi(k) = \begin{cases} \Phi_{\text{I}}(k), & 0 \le k \le k^{\text{crit}}; \\ \Phi_{\text{II}}(k), & k^{\text{crit}} \le k \le k^{\max}. \end{cases} \tag{2}$$

Eqn. (2) refers to the *physical* FD in the absence of any driving constraints. Let $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ denote the vector of parameters that constrain the vehicle speed, flow, and density, respectively, where $0 \le \theta_1, \theta_2, \theta_3 \le 1$. Expanding the idea presented in [10] we denote the *constrained* FD by $\tilde{\Phi}(k \mid \boldsymbol{\theta})$, which can be formulated as

$$\tilde{\Phi}(k \mid \boldsymbol{\theta}) = \min\left\{\theta_1 \gamma^{\max} k, \theta_2 \Phi\left(\frac{k}{\theta_3}\right)\right\}. \tag{3}$$

If $\boldsymbol{\theta} = (1, 1, 1)$ then $\tilde{\Phi}(k \mid \boldsymbol{\theta}) = \Phi(k)$ and no constraints are imposed. Using $\boldsymbol{\theta}$ several dynamic traffic management measures can be simulated. If a maximum speed of $\sigma^{\max}$ is imposed on the link, then $\theta_1 = \min\left\{\sigma^{\max} / \gamma^{\max}, 1\right\}$, while if one lane of a two-lane road segment is closed then $\theta_2 = \theta_3 = \frac{1}{2}$ (assuming that the the lane closure affects flows and densities proportionally).

Figure 1 illustrates different shapes of the constrained FD. The physical FD (with a quadratic hypocritical branch and a linear hypercritical branch for illustration purposes) is the same in all cases, while the constrained FD shown at the bottom varies depending on imposed speed, flow, and density constraints.

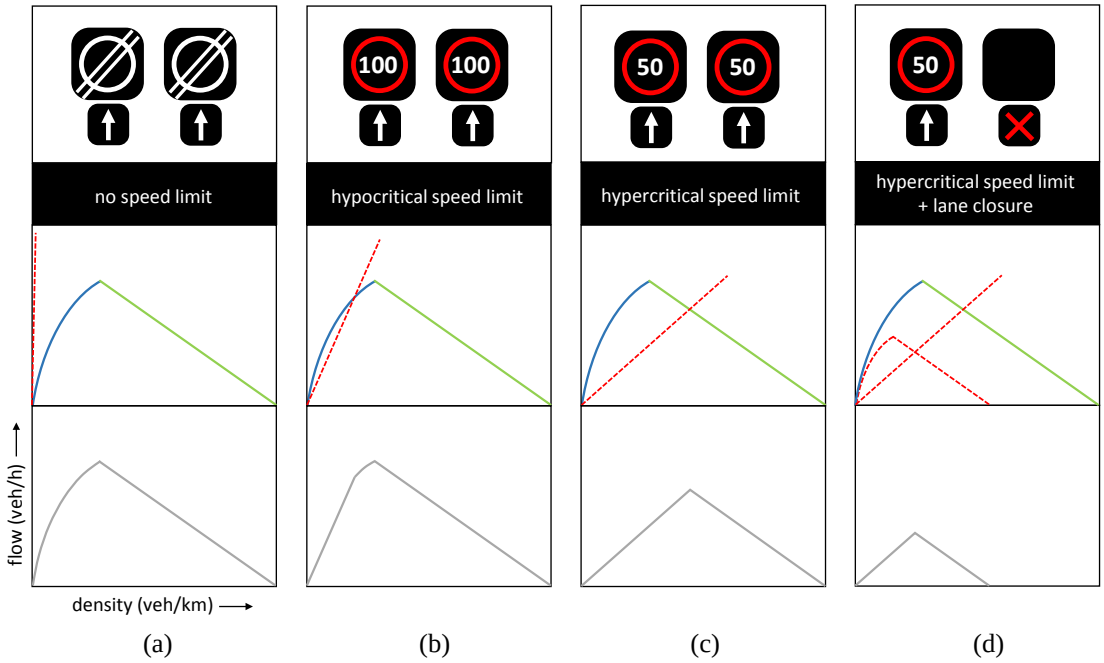


Figure 1 Constrained fundamental diagrams in case of (a) no speed limit, (b) hypocritical speed limit, (c) hypercritical speed limit, (d) hypercritical speed limit and lane closure.

3 Transitions between fundamental diagrams

We are interested in determining link outflow rates when there is a change in the FD starting at the upstream link boundary. Note that information about the FD (dynamic speed limits, dynamic lane management) only propagates downstream (with the driver of the vehicle), never upstream.

In the case of a fixed FD, an instantaneous flow rate increase at the link entrance may result in an acceleration fan in which traffic states at the downstream link boundary follow the shape of the FD [4]. Similarlarly, when we allow the FD to change shape, we consider changes in traffic states due to the transition from one (constrained) FD to the other. Such transitions between FDs occur conditional on the vehicle speed with which this information propagates.

Consider traffic state D in Figure 2(a). When there is a decrease in the maximum speed and the number of lanes available, then the traffic state changes to A (with the same flow rate but at a lower speed). This results in a temporary traffic state E with zero flow exiting the link, see Figure 2(b). Now consider traffic state A. When there is an increase in the maximum speed and the number of lanes available, then the traffic state eventually changes to D (with the same flow rate but at a high speed). This change is not instantanous but happens gradually via traffic states B and C and all traffic states in between. Practically, this can be simplified by only considering a subset of traffic states (e.g., B and C).

These (counter) clockwise transitions along the fundamental diagrams can be implemented as additional events in the algorithm proposed in [9]. In this solution method, the more traditional ‘flow rate change’ events propagate with the wave speed, while ‘route choice’ events propagate with the

vehicle speed. We now propose to add 'fundamental diagram change' events that also propagate with the vehicle speed, albeit with a very different impact. Case studies on networks will be presented in the full paper.

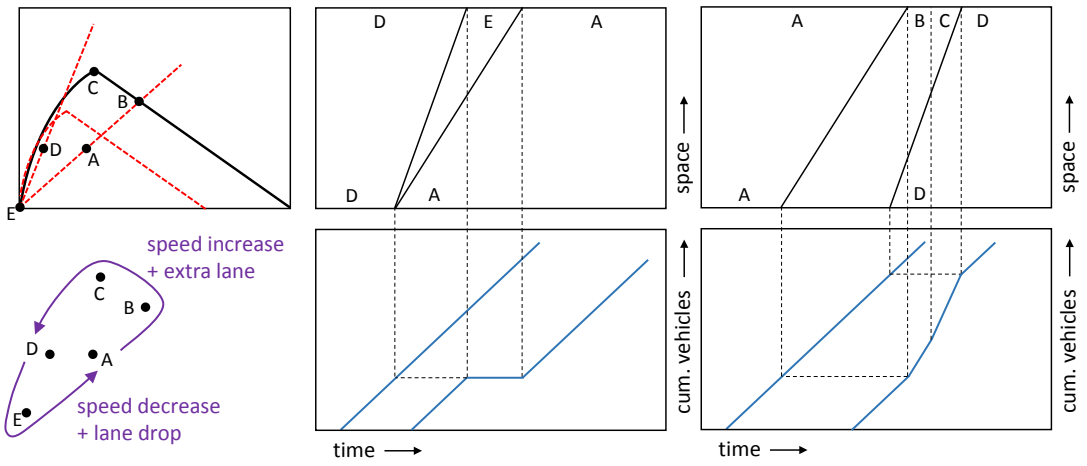


Figure 2 Constrained fundamental diagrams in case of (a) no speed limit, (b) hypocritical speed limit, (c) hypercritical speed limit, (d) hypercritical speed limit and lane closure.

References

[1] I. Yperman, *The Link Transmission Model for Dynamic Network Loading*, PhD Thesis, Katholieke Universiteit Leuven, Belgium, 2007.

[2] G. Gentile, "The general link transmission model for dynamic network loading and a comparison with the DUE algorithm", in: C.M.J. Tampère, F. Viti and L. Immers (eds.), *New Developments in Transport Planning: Advances in Dynamic Traffic Assignment*. Edward Elgar, Northampton, MA , USA , 153-178, 2010.

[3] J.P.T. van der Gun, A.J. Pel and B. van Arem, "Extending the Link Transmission Model with Non-Triangular Fundamental Diagrams and Capacity Drops", *Transportation Research Part B* 98, 154-178, 2017.

[4] M.C.J. Bliemer and M.P.H. Raadsen, "Continuous-Time General Link Transmission Model with Simplified Fanning, Part I: Theory and Link Model Formulation", *Transportation Research Part B*, in press.

[5] R.C. Carlson, I. Papamichail, M. Papageorgiou and A. Messmer, "Optimal mainstream traffic flow control of large-scale motorway networks", *Transportation Research Part C* 18, 193-212, 2010.

[6] M. Hajiahmadi, R. Corthout, C. Tampère, B. De Schutter and H. Hellendoorn, "Variable Speed Limit Control Based on Extended Link Transmission Model", *Transportation Research Record* 2390, 11-19, 2013.

[7] J.P.T. van der Gun, A.J. Pel and B. van Arem, "The Link Transmission Model with Variable Fundamental Diagrams and Initial Conditions", Transportmetrica B, in press.

[8] M.P.H. Raadsen, M.C.J. Bliemer and M.G.H. Bell, “An Efficient and Exact Event-Based Algorithm for Solving Simplified First Order Dynamic Network Loading Problems”, *Transportation Research Part B* 92, 191-210, 2016.

[9] M.P.H. Raadsen and M.C.J. Bliemer, “Continuous-Time General Link Transmission Model with Simplified Fanning, Part II: Event-Based Algorithm for Networks”, *Transportation Research Part B*, in press.

[10] M. Papageorgiou, E. Kosmatopoulos and I. Papamichail, “Effects of Variable Speed Limits on Motorway Traffic Flow”, *Transportation Research Record* 2047, 37-48, 2008.

Decision-Based Scenario Clustering for Decision-Making Under Uncertainty: applications in transport planning

Janosch Ortmann* and **Walter Rei**
Department of Management and Technology
UQAM School of Management
Université du Québec à Montréal

Michael Hewitt
Information Systems and Supply Chain Management Department
Quinlan School of Business
Loyola University Chicago
*Email: ortmann.janosch@uqam.ca

1 Introduction

Many problems related to transport can be formulated in terms of optimisation: some quantity is to be maximised or minimised subject to some constraints which are only partially known in advance. Consider for example stochastic network design, where a transport network must be built before the demands of goods to be transported are known, or fleet planning, where the number of vehicles to be used must be chosen under uncertainty.

In stochastic programming scenarios are used to approximate the distributions of the unknown parameters and formulate and solve multi-stage stochastic optimization models [1]. A general two-stage optimisation problem can be formulated as follows. Let $\mathcal{S}$ be a finite subset of a configuration space of parameters. These possible configurations $s \in \mathcal{S}$, as well as the probabilities p_s assigned to them, can either be inferred from past data that is used to derive probabilistic information, or generated by subjective analysis. We refer to elements of $\mathcal{S}$ as *scenarios*: each scenario corresponds to a possible configuration of parameters that might occur. In this terminology, the stochastic optimisation problem can be stated as

$$\inf_{y \in \mathcal{Y}} \sum_{s \in \mathcal{S}} p_s g(y, s), \tag{1}$$

where $\mathcal{Y}$ is a given set of decision variables (corresponding to the first-stage variables) and

$$g(y, s) = \inf_{x \in \mathcal{X}} h(x, y, s) \tag{2}$$

for $\mathcal{X}$ another given set of decision variables (corresponding to the second-stage variables).

In order to illustrate the general problem formulation, we give two examples:

Example 1.1 (Stochastic network design)**.** When designing a transport network, two types of decisions must be taken [2]: first, one chooses the structure of the network (*design decisions*) and secondly how to use this network to perform the operational activities considered (*flow decisions*). In our context, the design decisions must be taken before these stochastic parameters are known, while the flow decisions are taken once the realisations of the parameters have been revealed.

More precisely, consider a directed graph $G = (N, A)$ and a set of commodities K. For each scenario $s \in \mathcal{S}$, the stochastic parameters are given by the number of units of commodity $k \in K$ to be transported to vertex $i \in N$, and the capacity of each edge.

For each edge $e \in A$, the design decision corresponds to choosing whether to open e (at a fixed cost) or not. This choice is represented by the decision variable $y_{ij} \in \{0, 1\}$, and $\mathcal{Y}$ is the set of all decision variables y_e.

After the scenario s that actually occurs is revealed, the quantity x_e^{ks} of commodity $k \in K$ to be transported across edge e is chosen (at a variable cost per unit transported), and $\mathcal{X}$ is the set of all flow decision variables x_e^{ks}.

The optimisation problem corresponding to (1) now lies in choosing variables in $\mathcal{X}$ and $\mathcal{Y}$ such that the demands and edge capacities are satisfied, while minimising the total (fixed and per-unit) cost.

Example 1.2 (Biweekly fleet planning)**.** Our second test problem concerns the fleet-sizing problem faced by a freight carrier over a two-week horizon where the loads for the first week are known. The decisions are for the first week are the number of vehicles available at each terminal and the number of vehicles moving between each origin and destination. The decisions for the second week are similar, but the vehicle supply at each terminal is determined by the decisions of the first week. If the vehicle supply of a location at the end of the second week is different than what it was at the beginning of the first week, then a penalty is incurred. Here the set $\mathcal{Y}$ of first-stage variables is given by the vehicle movements in the first week, while the second stage variables in $\mathcal{X}$ correspond to vehicle movements in the second week.

In order to accurately model the underlying sources of uncertainty, a large number of scenarios may need to be generated, which leads to high computational complexity and may even render the problem numerically intractable to solve. In this situation, it may be tempting to replace the stochastic parameters with point estimates, such as their expected value. However, this approach

leads to errors which are difficult to estimate, and the optimal solution to (1) may have characteristics that neither any of the deterministic solutions nor the expected value solution exhibit [1].

In the proposed talk we will introduce a general methodology that permits control over both computational complexity and the error incurred and show how this methodology is applied to the two example transport problems described above. We introduce a decision-based distance on the set of scenarios, according to which scenarios s_1 and s_2 are close if and only if the optimal solution associated to scenario s_1 is close to optimal for scenario s_2. This distance induces a natural weighted graph structure on the scenarios and we can use graph clustering methods to identify structure in the scenario space associated with decision-making contexts under uncertainty. By identifying clusters of scenarios with mutually acceptable decisions, this leads to new bounds and solution heuristics in transport applications such as stochastic network design and fleet planning problems.

2 Methodology

Suppose that we had an oracle that predicts with certainty which scenario s will occur. Then we could always choose the best decision

$$y_s^* = \arg\min_y g(y; s) \tag{3}$$

under this scenario. In reality, a perfect oracle does not exist, and this process will lead to incorrect predictions and therefore non-optimal decision being taken from time to time. In order to quantify this error, we introduce the *opportunity cost* of taking the decision associated to scenario s_1 when another scenario s_2 actually occurs. Denote this cost by $\delta(s_1 | s_2)$. Since $\delta(s_1 | s_2) \neq \delta(s_2 | s_1)$ in general, we will symmetrise and define the *opportunity cost distance function* on $\mathcal{S}$ by

$$d(s_1, s_2) = \delta(s_1 | s_2) + \delta(s_2 | s_1), \qquad s_1, s_2 \in \mathcal{S}. \tag{4}$$

This function introduces a notion of distance on the set $\mathcal{S}$ of scenarios, which enables us to compare scenarios on a decisional basis. It is natural to now identify groups of scenarios which are close to each other with respect to this distance, since the decision associated to one scenario in this group will still be close to optimal for the others. Such groups also yield another way of estimating the risk associated to the predictions made by the oracle.

Mathematically, our technique of finding such clusters consists of defining a graph (called *affinity graph*)with vertex set $\mathcal{S}$, based on the notion of distance induced by d: the smaller the opportunity cost between two scenarios s_1, s_2, the shorter the length of the edge between s_1 and s_2. We then apply graph clustering techniques such as Ncut [4] or its relaxation, spectral clustering [4, 3, 5] in

order to identify groups of scenarios which are close to each other with respect to the opportunity cost distance, that is scenarios with a jointly acceptable solution to the optimisation problem (1).

3 Applications

We will also outline how the methodology described above can be applied to the two stochastic optimisation problems from Examples 1.1 and 1.2. The grouping of scenarios in the decision space leads to a better understanding of potential compromise solutions, which in itself leads to a better grasp of the problem and potential solution approaches. Moreover, we obtain new upper and lower bounds and an analysis of the parameters in terms of the decision-based clusters.

By re-defining the expected value of perfect information (EVPI) bound [1] to the clusters, we obtain new lower bounds. By solving reduced-sized stochastic models based on aggregating the scenarios in a cluster, or choosing a representative scenario from each cluster, we can obtain efficient heuristic approximations to the full problem, as well as natural upper bounds. We can also apply the meta-heuristic explained in [2] to our clustering.

Finally, the grouping permit a decision-based analysis of the input parameters to the problem. By analysing the distribution of the parameters across the clusters, we can determine how strongly and in which way each parameter influences the optimal solution to the problem. We can also identify edge cases at which the optimal decision changes.

References

[1] Birge, J. R., and Louveaux, F. V. *Introduction to Stochastic Programming*, second ed. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, 2011.

[2] Crainic, T., Hewitt, M., and Rei, W. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research 43* (2014), 90 – 99.

[3] Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems* (2002), T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, MIT Press, pp. 849 – 856.

[4] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888 – 905.

[5] von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing 17*, 4 (2007), 395 – 416.

A Large-scale Neighborhood Search Approach to Airport Slot Allocation

Nuno Antunes Ribeiro

CITTA – Department of Civil Engineering

University of Coimbra

Email: nribeiro@student.uc.pt

Alexandre Jacquillat

Heinz College

Carnegie Mellon University

António Pais Antunes

CITTA – Department of Civil Engineering

University of Coimbra

1 Introduction

Air traffic demand has grown to exceed available capacity at many airports worldwide, resulting in the routine occurrence of flight delays and high costs to airports, airlines and passengers. For instance, the nationwide impact of air traffic congestion in the United States was estimated at over $30 billion in 2007 [1]. Absent opportunities to expand airport capacity, it is necessary to resort to demand management to prevent over-capacity scheduling. Demand management involves administrative rules or economic incentives to limit the number of flights scheduled at busy airports and at busy times by rescheduling flights over the day and, in some cases, by reducing the total number of flights. The vast majority of busy airports outside the United States are subject to schedule coordination, operated under the aegis of the International Air Transport Association (IATA). In 2017, schedule coordination was applied at 177 airports, serving a total of 3.15 billion passengers annually. Under this process, the airports provide a value of *declared capacity*, which determines the number of *slots* available to allocate to the airlines per hour. For each season, the airlines submit their slot requests to a *slot coordinator*, which performs slot allocation according to the Worldwide Slot Guidelines (WSG) set forth by IATA [2]. These guidelines specify rules and priorities that create coupling constraints across the allocation of slots at multiple times of day and on multiple days of year. As a result, slot allocation is a highly complex combinatorial problem, which carries enormous weight for airlines, airports, and passengers. Optimization models have been proposed to support slot allocation decisions at schedule-coordinated airports. These models primarily aim to minimize the deviations of the schedule of flights from the airlines' requests. They have shown considerable promise to improve slot allocation outcomes. However, existing optimization approaches remain limited to small- and medium-size airports due to the combinatorial

complexity of slot allocation. In contrast, the implementation of slot allocation optimization models at large-size airports remains intractable.

We address this issue by developing an original optimization approach to solve the slot allocation problem at the largest schedule-coordinated airports. We formulate an integer programming model that captures all the rules and priorities from IATA's Worldwide Slot Guidelines, and we develop a new algorithm based on large-scale neighborhood search to solve it efficiently at the busiest airports. The proposed algorithm starts by generating a feasible slot allocation solution, and then improves it iteratively by re-optimizing slot allocation decisions for a subset of slot requests. The algorithm is implemented at the Lisbon Portela Airport (LIS), one of the top-20 busiest airports in Europe. Results show that it provides optimal or near-optimal solutions in 6-10 hours of computation in settings where commercial solvers fail to identify the optimal solution after 7 days of computation. Thus, this work considerably enhances the capabilities of slot allocation models and algorithms.

2 The Priority-based Slot Allocation Model (PSAM)

The slot allocation problem can be viewed as an optimization problem that can be stated, in general terms, as follows: "given a set of airline requests for slots during a season of operations and a set of constraints resulting from the airport's declared capacities and the IATA guidelines, propose a combination of slot assignments (i.e., a "slot allocation") that minimizes the difference between the proposed schedule of flights and the schedule that would have resulted from the airlines' requests in the absence of capacity constraints". This problem is formulated in Ribeiro et. al. 2018 [3] as an integer programming model, named Priority-based Slot Allocation Model (PSAM). In this work we extend PSAM to consider additional requirements of the slot allocation process, such apron and terminal capacity constraints.

Qualitatively, PSAM can be formulated as follows:

Minimize	Schedule displacement	(1)
subject to	Capacity constraints	(2)
	Flight connection constraints	(3)
	Schedule regularity constraints	(4)
	IATA priority constraints	(5)

The objective is to minimize an aggregate measure of schedule displacement, typically measured by the total displacement. To ensure the solution feasibility, a set of constraints need to be considered: (i) *capacity constraints*, which ensure that the declared capacities of the airport are never exceeded. (ii) *connectivity constraints*, which ensure the minimum turnaround times between consecutive flights operated by the same aircraft. (iii), *schedule regularity constraints*, which state that flights belonging to the same slot request must be allocated to the same time of the day on each day of the season. For instance, if a slot is requested on Mondays

and Wednesdays for 15 weeks, the flights need to be scheduled at the same time of the day on the corresponding 30 days. Note that this requirement creates coupling constraints across all days in the season, thereby considerably increasing the complexity of the slot allocation problem. (iv) *IATA priority constraints*, which ensure that the allocation of slots is performed sequentially given the four priority classes specified into the IATA's Worldwide Slot Guidelines, specifically: (i) "historic slots" (i.e., slots owned by the same airline in the previous equivalent season that were used at least 80% of the time); (ii) "change-to-historic slots" (i.e., historic slots for which the airline requests a change such as re-timing or the use of another aircraft); (iii) "new-entrant slots" (i.e., slots requested by airlines owning less than five slots a day); and (iv) "other slots" (i.e., slot requests that do not belong to any of the three other priority classes).

3 Heuristic Approach to Slot Allocation

We propose a scalable algorithm based on large-scale neighborhood search to solve the slot allocation problem. The goal of our algorithm is to derive optimal, or near-optimal, where direct implementation using commercial solvers of optimization remains intractable. The proposed algorithm relies on the following logic. In general terms, there exists a "limit" for solving the slot allocation problem with commercial solvers. One of the main determinants of this limit is, of course, the number of slot requests. However, there is no one-to-one relationship between size and computational performance; for instance, all else being equal, the more significant the imbalances between slot demand and airport capacity, the more computation effort is required to solve PSAM. Therefore, we subdivide the full set of slots into smaller subsets based on the size of the problem and other factors (e.g., demand-capacity imbalances).

Our algorithm involves a *constructive heuristic* and an *improvement heuristic* (shown in Figure 1). The *constructive heuristic* aims to find an initial feasible solution to the slot allocation problem by dividing the set of slot requests into smaller groups by decreasing order of priority (i.e. change-to-historic, new-entrants, and other slots) and frequency. Thus, for each priority class, the constructive heuristic allocates, first, the slots requested for the full season, then those requested on most weeks of the season, etc. Then, the improvement heuristic iteratively improves this solution using a "destroy and repair" approach. At each iteration, it removes a subset of slot requests from the assignment determined by the latest solution and solves the PSAM for the remaining slot requests. The full set of slot requests is still included in the model to ensure global feasibility, rather than local feasibility. However, only a subset of all slot requests are re-allocated. In other words, the improvement heuristic explores the full solution space iteratively by decomposing the slot allocation problem into smaller sub-

problems, fixing many decision variables to their previous values, and performing local optimization at each iteration.

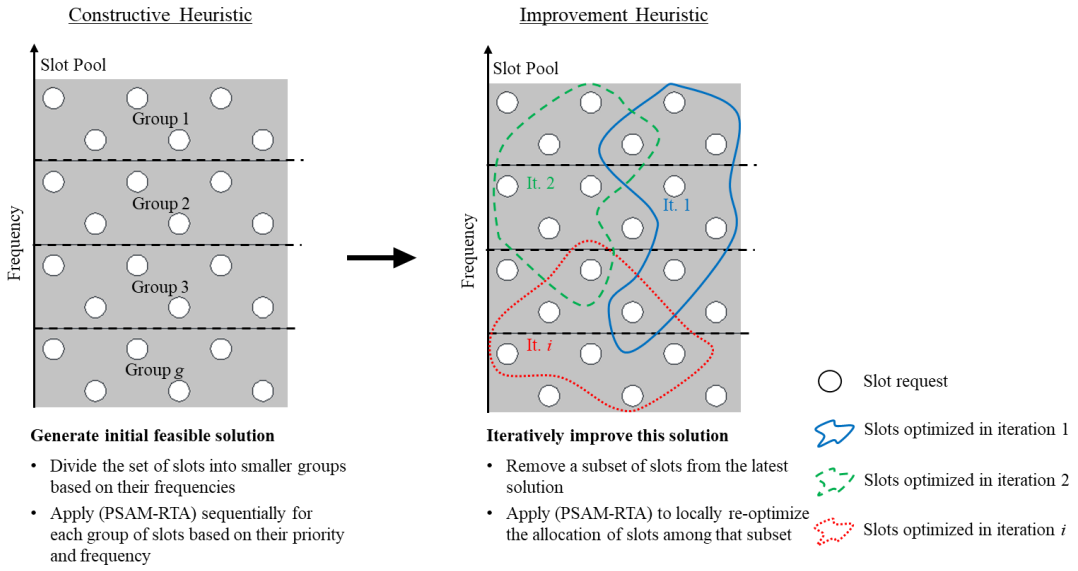


Figure 1 – Schematic representation of the heuristic proposed

Computational results using real-world data from the Lisbon Airport suggest that optimal or near-optimal solutions to PSAM can be obtained in reasonable runtimes. Specifically, while direct implementation of PSAM with commercial solvers yields a solution within 5-10% of the optimum after 2 days and within 0.5-2% of the optimum after 7 days, the proposed algorithm provides a solution within 2-5% of the optimum after 30 minutes and within 0-0.03% of the optimum after 10 hours. Extensive sensitivity analyses also showed that the algorithm performs better than more straightforward implementations of large-scale neighborhood search methods in this context, and that results are robust to a number of calibration parameters. Ultimately, this work augments the capabilities of slot allocation models and algorithms. Its application in support of slot allocation at major schedule-coordinated airports worldwide can result in flight schedules that match airlines' slot requests and passenger demand more effectively than existing approaches based on specialized software and ad hoc allocation decisions.

References

[1] Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., & Zou, B. (2010). Total Delay Impact Study. Technical report, National Center of Excellence for Aviation Operations Research, College Park, MD.

[2] IATA (2018). Worldwide Slot Guidelines, 8th Edition. Montreal, Canada.

[3] Ribeiro, N.A., Jacquillat, A., Antunes, A.P., Odoni, A. R., & Pita, J. P. (2018). An optimization approach for airport slot allocation under IATA guidelines. Transportation Research Part B: Methodological, 112, 132-156.

Improving pedestrian dynamics by preventing counter-flow

Nicholas Molyneaux

Riccardo Scarinci

Michel Bierlaire

Transport and mobility laboratory

École polytechnique fédérale de Lausanne

Email: nicholas.molyneaux@epfl.ch

14 October 2018

1 Introduction

Exploiting the full capacity of pedestrian infrastructures is vital to ensure a satisfactory level-of-service and limit costs. This aspect is true for many different infrastructure types such as transportation hubs, metro stations, conference centers or even open streets. As pedestrian dynamics contain high spatial and temporal variability, dimensioning structures for the peak demand can be very costly and require significant space, whilst the structure is only used to capacity for very short periods of time. Therefore, to prevent congestion and its negative effects we propose a traffic control strategy for pedestrian traffic which is integrated in a dynamic traffic management system.

Multiple control strategies exist for road traffic, such as signalized intersections [1], ramp metering [2] or variable message signs [3]. Their effectiveness has be proven in real world applications and simulation laboratories [4, 5]. We think that similar control strategies could improve the pedestrian flows. This requires the inclusion of the pedestrian traffic specificities and tailored control strategies.

As experienced by many individuals and shown in studies [6], counter flow in pedestrian traffic is responsible for a significant increase in travel time. This happens as people have to "slalom" between the people coming in the opposite direction. In order to prevent this, we propose a control strategy for preventing counter flow in corridors: flow separators. Counter flow can be prevented by splitting the corridors dynamically based on the pedestrian flows coming in each direction. This strategy is included in a Dynamic Traffic Management System (DTMS) framework designed for pedestrians.

2 Methodology

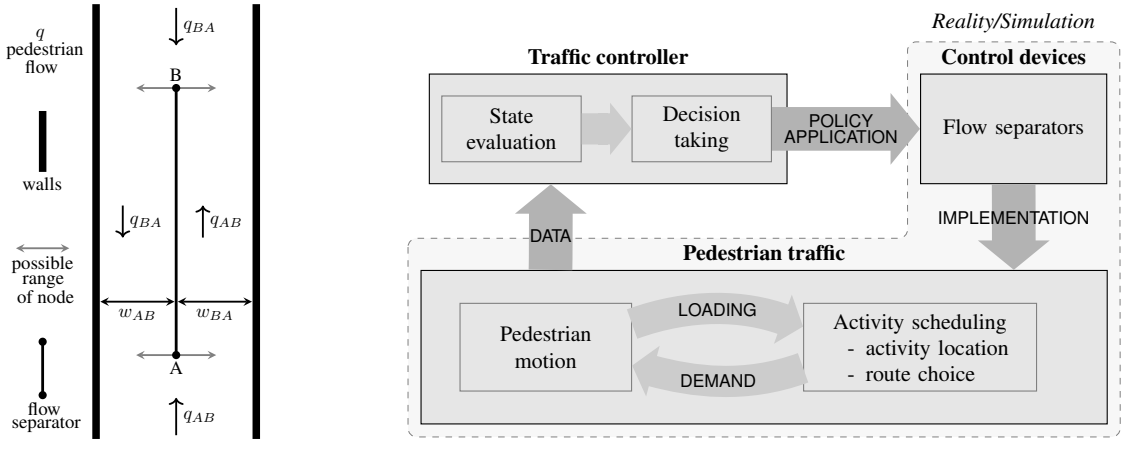
The current approach divides the corridor dynamically based on the measured flows. Figure 1a presents a schematic setup where a flow separator is installed. The flows are measured at the entrance of each dedicated side and the width allowed per direction is then proportional to these flows using the following equation:

$$w_{AB}(t) = \begin{cases} w_{AB}^{min}, & \text{if } w \cdot \dfrac{q_{AB}}{q_{AB}+q_{BA}} \leq w_{AB}^{min} \\ w_{AB}^{max}, & \text{if } w \cdot \dfrac{q_{AB}}{q_{AB}+q_{BA}} \geq w_{AB}^{max} \\ w \cdot \dfrac{q_{AB}}{q_{AB}+q_{BA}}, & \text{otherwise} \end{cases} \tag{1}$$

where w is the total width of the corridor, w_{AB} is the width of the corridor from A to B, q_{AB} the pedestrian flow entering at A, w_{AB}^{min} and w_{AB}^{max} the minimum (resp. maximum) width dedicated to the direction AB. The width of the corridor from B to A is naturally the remainder of the corridor width.

This approach has the advantage of requiring no calibration. The only parameters which must be fixed are the minimal and maximal widths. These should correspond to the width required by one person to walk freely, without being hindered by the walls [7].

This strategy is integrated in a DTMS specifically designed for pedestrians (Figure 1b). The DTMS is composed of the pedestrian traffic module, the traffic controller which acts as a brain and the control devices which apply the decisions taken be the controller.



(a) Flow separator setup.

(b) DTMS framework for pedestrian traffic, from [8].

Figure 1: Schematic setup for the flow separator which is included inside the dynamic traffic management system for pedestrians.

3 Results

The flow separator is tested in a simulation laboratory [4] which reproduces the effect of a DTMS. The pedestrian traffic is simulated using the NOMAD microscopic simulator [9] and the route choice is modeled as the shortest path algorithm. For details about this DTMS for pedestrians, we refer to [8]. A straight corridor with variable demand is considered. The demand pattern follows two sine-functions with a shift in phase between them.

As seen in Figure 2, separating flows by direction is very efficient for preventing increase in travel time. The mean travel time of all pedestrians decreases from 38.02[s] to 30.19[s], alongside the travel time variance which goes from 10.22[s] to 3.29[s].

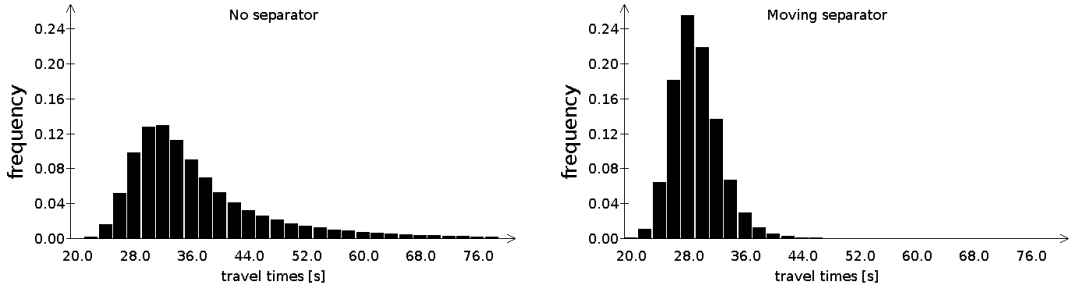


Figure 2: Distribution of the pedestrian's travel times for two scenarios: without (left) and with (right) flow separators. The flow separators significantly decrease the travel time.

4 Conclusion & future work

The proposed flow separator control strategy is very effective at preventing the increase in travel time due to counter flow. Not only is the mean travel decreased, but the variance also significantly decreases as well. Although only a single corridor has been used, the benefits of separating pedestrian flows by direction is not limited to simple infrastructure. This approach can be used in many different category of infrastructure to improve the pedestrian dynamics.

The next steps involve two major aspects. Firstly, an extension of the case study to cover part of a real train station (Lausanne, Switzerland). In this way, multiple flow separators can be considered. This can also allow for coordination between the different elements. Secondly, transforming the control strategy to become anticipative and not only reactive. By integrating a model-predictive-control component into the strategy, the width dedicated to each direction can be allocated based on the expected flows, not only the current flows. This is particularly important for train stations, where the flows can be accurately predicted based on the train timetable.

Acknowledgments

This research was performed as part of the TRANS-FORM (Smart transfers through unravelling urban form and travel flow dynamics) project funded by the Swiss Federal Office of Energy SFOE and Federal Office of Transport FOT grant agreement SI/501438-01 as part of JPI Urban Europe ERA-NET Cofound Smart Cities and Communities initiative. We thankfully acknowledge both agencies for their financial support.

References

[1] A. Di Febbraro, D. Giglio, and N. Sacco. Urban traffic control structure based on hybrid petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):224–237, Dec 2004.

[2] Markos Papageorgiou, Habib Hadj-Salem, and Jean-Marc Blosseville. Alinea: A local feedback control law for on-ramp metering. *Transportation Research Record*, 1320(1):58–67, 1991.

[3] M Wardman, P.W Bonsall, and J.D Shires. Driver response to variable message signs: a stated preference investigation. *Transportation Research Part C: Emerging Technologies*, 5(6):389 – 405, 1997.

[4] Moshe Ben-Akiva, David Cuneo, Masroor Hasan, Mithilesh Jha, and Qi Yang. Evaluation of freeway control using a microscopic simulation laboratory. *Transportation Research Part C: Emerging Technologies*, 11(1):29 – 50, 2003.

[5] Hani S. Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3):267–292, Sep 2001.

[6] C Burstedde, K Klauck, A Schadschneider, and J Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507 – 525, 2001.

[7] Ulrich Weidmann. Transporttechnik der fußgänger: transporttechnische eigenschaften des fußgängerverkehrs, literaturauswertung. *IVT Schriftenreihe*, 90, 1993.

[8] Nicholas Molyneaux, Riccardo Scarinci, and Michel Bierlaire. Two management strategies for improving passenger transfer experience in train stations. In *Proceedings of the 18th Swiss Transport Research Conference*, Ascona, Switzerland, 2018.

[9] Mario Carlos Campanella. *Microscopic modelling of walking behaviour*. TRAIL, 2016.

A General Theory of Access: Extended Abstract

David Levinson

TransportLab, School of Civil Engineering

University of Sydney

Email: david.levinson@sydney.edu.au

1 Introduction

Accessibility (A), the ease of reaching destinations has numerous dimensions. The key dimensions are listed below.

- Where (i)
- When (h)
- Why (z)
- Mode (m)
- Who (p)
- Cost (c)
- Measurement (t,θ,N,R)

This paper develops a consistent formulation of access that accounts for accessibility's multiple dimensions. By doing so, it reveals unexplored territory in accessibility space that should be valuable to those using accessibility as a performance measure, including transport engineers, planners, geographers, and policy-makers. Measures of accessibility, and comparisons across measurements are presented in the full paper.

2 Where

We start with *where*, (i). Where is the accessibility being measured? We divide space into smaller units of geography (at the limit, every point; more practically, every parcel or street block or Census block group, or transport analysis zone), which we denote as (i). We them measure A_i. But we might also be interested in a more aggregate measure, we can average all of the accessibilities

measured at all of the i's, and develop a system-wide average. But how you average matters, so we often *person-weight* the average, so the access at i is weighted by (multiplied by) the population in i (S_i), and the whole thing is divided by the total population in the area of interest ($S_.$), to give the population-weighted average, which we denote as $A_.$.

3 When

Next we can consider *when*, (h). For instance the accessibility in the peak hour may differ from that at 4:00 am. There are two reasons.

First, the network differs at 4:00 am from that at 8:00 am, for roads there is less traffic, but for transit, there is less service and so more waiting or access/egress time.

Second, the opportunities ($O_{i,h}$) differ by time of day, stores are open or closed, jobs are available or unavailable depending on time of day.

But accessibility at 8:00 am differs from 8:01 as well. Traffic differs somewhat due to the ebb and flow of congestion and shockwaves. Transit varies more systematically: at 8:01, the scheduled 8:00 am bus or train may have just left, increasing the waiting time at the stop or station, which implies the transit travel time differs greatly. Averaging transit travel times across a peak hour can provide a more realistic measure than just a sample at a single point in time.We may do a simple average, or something more complicated.

4 Why

Examining *why we travel* (or *what is being accessed*) (z) brings to the fore the question of what kind of opportunities are of interest. They may be jobs, or houses, or the number of jobs available after controlling for workers, or stores, and so on.

Historically, when measuring access to jobs, analysts have considered the existence of jobs, and measured them as if they are interchangeable (though noting they of course are not). Some have stratified jobs by income or by type to examine the number of jobs available to individuals with specific skills. This analysis is very dependent upon data availability. Also not all jobs are available at the same time, cleaning crews tend to operate outside of regular business hours, so this relates back to the question of *when*.

Perceived opportunities differs from objectively measured opportunities.

5 Mode

The dimension of *how we travel* (or *by what mode*) (m) indexes accessibility by whether the trip is made on foot, by bike, by public transport, or by automobile (or by any number of numerous

other modes we might consider). Clearly speeds vary by mode. The perceived speed differs from the objectively measured speed.

6 For Whom

The question of accessibility for *whom* (p) relates to where. At the limit, looking at individual persons and highly localized places (i) will be identical, at least at the points in time where the individual occupies that place. Keep in mind aggregating spatially (where the subgroup is people who are adjacent to each other at a point in time, e.g. residents of a block) is a very special case of aggregating by groups generally. In the case of equity, we might be interested in places, but we have interest in other kinds of groups. We are interested not in individuals as such, but in subgroups, for instance minorities, and ask how their accessibility compares with other subgroups, or the population at large.

7 Incidence: Who Bears Which Cost

Initially the question of *how much* (c) was the cost between i and j was taken to be a distance, modeled on Newton's Laws of Gravity. Later travel time was used. But in social analyses, private individuals travel time is only part of the cost for making a trip, the monetary costs paid by the traveler, and the subsidy of those costs paid by society also occur. But even more, there are externalities, like pollution, crashes, and congestion imposed on others which might be considered.

7.1 Perceived Time

The time that is used in accessibility is often assumed to be objectively measured time, but we may want to think about it as user perceived time. Time perception varies with conditions under which time is experienced, and while it varies from person to person, there are conditions which most people will perceive to be longer (waiting, stop and go traffic) and conditions which will be perceived to be shorter (moving unimpeded) than actual.

While getting the perceived time for every origin-destination pair for every individual is likely to be impossible, models of perceived time can be estimated and used as part of the cost matrix (C_{ij}).

7.2 Quality of Time

Related to perceived time is the idea of the quality of time. People may accurately perceive time, but wish it were different. Handling this is more difficult. Modes vary in the quality of time. People are happier walking or riding a bike than riding a bus or crowded train. Within mode

choices models, for transit, for instance, the weighting on out-of-vehicle time is much greater than on in-vehicle time, and while some of this accounts for time perception, some also accounts for time quality.

By using satisfaction as a modifier to the travel cost, jobs could be discounted according to the degree of dissatisfaction associated to the trip by a given mode. For example, two jobs reachable with a trip satisfaction rating of 50% may be worth one job reachable with a satisfaction rating of 100%. This would enable the use of satisfaction-based accessibility in an easily-understood cumulative opportunities framework. Combining perceived (or reported) travel time (as distinct from objectively measured travel time using GPS) with satisfaction in accessibility measures is another direction for future research. The issue, which cannot be addressed with current data sets, is the extent that dissatisfaction already embeds higher perceived travel times, or the degree to which they are two distinct phenomena.

8 Formalization

In our formal definitions, the question of *how accessibility is assessed* (t,θ,N,U,R) is presented in separate sections, which define each measure mathematically. In this abstract, we present the first, a series of measures based on the initial, or primal, formulation. There are levels of distinctions within that. So for instance, in what we refer to as *primal* accessibility, we might use a travel time threshold (t) or an impedance function (f) with specific parameters (θ), all of which give different, though systematically related, numeric answers. But other families of methods (the Dual, Place Rank, Utility, etc.) provide far more differentiated results.

8.1 Hansen's Accessibility

Primal accessibility as presented here is a generalization of the first accessibility formuation by Hansen.[1] In the *primal* accessibility (A) problem, we solve for how many destinations (O) can be reached in t minutes from origin i? For completeness, here we note the subscript for time-of-day (h), for activity type or purpose (z), and for mode (m), considering particular set of costs (c) and population subgroup (p) (income category, racial group, modal availability, etc.), and recognize that we could add any other sub-categorization we may wish to impose. While this may seem pedantic, it also reveals the richness of the problem, which is multi-dimensional. We present a reduced version as well.

This primal measure of accessibility is foremost *positive*, measuring how many activities can be reached. One could, however, impose a *normative* standard, and insist that it should be above some number (N). It implies the question of whether providing such levels of accessibility is an

[1] (Hansen, 1959).

appropriate use of government. For some activities, most people probably agree that it is (a fire station should be within X minutes of anyone who lives in a city, or $X+Y$ minutes in a rural area), and for others it is not. There is no standard for the number of jobs reachable within 30 minutes, but all else equal, more is better than fewer. Discussions of the '30-minute city',[2] for instance, suggest that 75% of the people should be able to reach jobs within 30 minutes using transit. We develop the Primal in the most detail, showing how the different dimensions play out.

This definition bears on the distribution of jobs and housing as much as on transit service.

$$A_{i,h,z,m,c,t,p} = \sum_{j=1}^{J} O_{j,h,z} f(C_{ij,h,m,c}) \tag{1}$$

We might abbreviate this, dropping subscripts.

$$A_i = \sum_{j=1}^{J} O_j f(C_{ij}) \tag{2}$$

8.2 Cumulative opportunities, Isochronic

To apply this in practice, the function of costs needs to be specified. First we present the cumulative opportunities formulation.

$$f(C_{ij,h,m,c}) = \begin{cases} 1 & \text{if } C_{ij,h,m,c} < t \\ 0 & \text{if } C_{ij,h,m,c} \geq t \end{cases} \tag{3}$$

Or in abbreviated form

$$f(C_{ij}) = \begin{cases} 1 & \text{if } C_{ij} < t \\ 0 & \text{if } C_{ij} \geq t \end{cases} \tag{4}$$

8.3 Gravity, 'Potential,' Distance Decay, Time-weighting

Alternatively, one could easily use a time-weighted cumulative opportunities (gravity) formulation. One commonly used function is negative exponential. In this case we might write the accessibility as: $A_{i,h,z,m,c,\theta,p}$ to account for the use of the impedance factor (θ) rather than a time threshold (t). This is sometimes referred to as 'distance decay', though in practice it is really 'time decay'. It is also referred to 'Potential', though this term is used for many different definitions.

$$f(C_{ij,h,m,c}) = e^{\theta_{e,m} C_{ij,h,m,c}} \tag{5}$$

The impedance factor ($\theta_{e,m} < 0$) is measured empirically and varies with mode (m). Previous studies have found values on the order of -0.08.

There are numerous other commonly used impedance functions, with empirically estimated parameters, as shown in Table 1.

[2] (Greater Sydney Commission, 2018).

Table 1: Illustrative Distance Decay Functions, Applicable to any mode (m)

Decay Function	$f_{C_{ij}}$
Exponential	$exp(\theta_e * C_{ij})$
Gaussian	$exp(\theta_g * {C_{ij}}^2)$
Log-normal	$exp(\theta_l * ln^2 C_{ij})$
Exponential-square-root	$exp(\theta_{\sqrt{e}} * {C_{ij}}^{0.5})$
Gravity	${C_{ij}}^{-2}$
Potential	${C_{ij}}^{-1}$ **??**
Scaling	${C_{ij}}^{\theta_s}$
No Distance Decay	1

While C_{ij} is often taken to be the time-cost, it could be the distance cost, or any other cost.

8.4 Person-weighting

To get a system average, we might sum the accessibility in each origin, weighted by the number of people who live in that zone ($S_{i,.}$).

First we note the population in a zone is the sum of all the subgroups in the zone.

$$S_{i,.} = \sum_{p=1}^{P} S_{i,p} \tag{6}$$

$$A_{.,h,z,m,c,t,.} = \frac{\sum_{i=1}^{I} A_{i,h,z,m,c,t,p} S_{i,.}}{\sum_{i=1}^{I} S_{i,.}} \tag{7}$$

8.5 Active and Passive

We might think of Equation 2 as *active accessibility*.

Alternatively, considering how easy it is to be reached, rather than to reach, we have *passive accessibility*[3]:

$$A_i^{\circlearrowleft} = \sum_{j=1}^{J} O_j f(C_{ji}) \tag{8}$$

Passive accessibility has applications for retailers who want to know how easily customers can reach them, or employers seeking to fill jobs.

8.6 Relative

Relative accessibility measures how accessible a particular zone is compared to the region at large.

[3]The terms 'active' and 'passive' were used in: (Papa and Coppola, 2012; Cascetta, 2009).

For instance, it might be the fraction of person-weighted regional accessibility that is attained in a given zone (i).

$$A^{\ominus}_{i,h,z,m,c,t,.} = \frac{A_{i,h,z,m,c,t,.}}{A_{.,h,z,m,c,t,.}} \tag{9}$$

This averages 1.0, but can be higher are lower depending on whether the zone in question is higher or lower than the regional average.

Alternatively, it could be the fraction of the regions total jobs reachable in a time threshold (t) from zone (i).

$$A^{\oslash}_{i,h,z,m,c,t,.} = \frac{A_{i,h,z,m,c,t,.}}{\sum_{j=1}^{J} O_{j,h,z}} \tag{10}$$

One could construct a person-weighted average relative accessibility.

$$A^{\oplus}_{.,h,z,m,c,t,.} = \frac{\sum_{i=1}^{I} A^{\oslash}_{i,h,z,m,c,t,.} S_{i,.}}{\sum_{i=1}^{I} S_{i,.}} \tag{11}$$

8.7 Multiple Time Slices

The problems above are laid out as if all opportunities are available 24 hours a day. But stores and restaurants open and close. Jobs have hours when they are available. Transport services vary by time-of-day (from minute-to-minute and hour-to-hour). Congestion ebbs and flows. But the value of an opportunity, and thus overall access, depends on when it is experienced. We need to consider the cost of travel (including the travel time) at a given time-of-day (h) by mode (m) considering costs (c) ($C_{ij,h,m,c}$). We may sum over opportunities available at a given time-of-day ($O_{j,h,z}$) if we appropriately weight them. Solving separately for a given trip purpose (z) and mode (m) at a given time-of-day (h) we have:

Summarizing across the whole day gives:

$$A_{i,.,z,m,c,t,p} = \sum_{j=1}^{J} \sum_{h=1}^{H} O_{j,h,z} U_h f(C_{ij,h,m,c}) \tag{12}$$

where h indexes activity time-of-day, and U_h weights the value of each time slice.

We normalize activities so that:

$$\sum_{h=1}^{H} U_h = 1 \tag{13}$$

We might choose U_h to be the fraction of the time-of-day represented by a time slice (h). So if h were hours, $U_h = \frac{1}{24}$ and if h were minutes, $U_h = \frac{1}{1440}$. We could of course alternatively rank peak or waking hours higher.

Because transit accessibility is so potentially variable depending on schedules, we might choose U_h to the be the fraction of the (say, two-hour) peak period represented by time slice (h), which if h is small (e.g. one-minute, so $U_h = \frac{1}{120}$) would give us an approximately continuous accessibility measure.[4]

8.8 Multiple Activities

The problems above are laid out as if there were only one opportunity or activity type or purpose of interest, e.g. jobs. But the value of a location, and thus overall access, depends on many different types of opportunities. We may sum over opportunities if we appropriately weight them:

$$A_{i,h,.,m,c,t,p} = \sum_{j=1}^{J} \sum_{z=1}^{Z} O_{j,h,z} W_z f(C_{ij,m,c}) \tag{14}$$

where z indexes activity types, and W_z weights the value of each activity type.

We normalize activities so that:

$$\sum_{z=1}^{Z} W_z = 1 \tag{15}$$

We might choose W_z to be the average share of time per day spent at each activity, or some other indicator of its importance. This could of course be defined uniquely for each individual if the data were available.

8.9 Multiple Time Slices and Multiple Activities

We can combine the notion of multiple time slices and multiple activities, for each given mode (m) considering cost components (c).

$$A_{i,.,.,m,c,t,p} = \sum_{j=1}^{J} \sum_{z=1}^{Z} \sum_{h=1}^{H} O_{j,z,h} W_z U_h f(C_{ij,h,m,c}) \tag{16}$$

8.10 Multiple Time Slices, Multiple Activities, Multiple Modes

Thus far, we have explicitly solved this problem for each given mode (m).

Combining the modes is trickier. While activity and time-of-day affect opportunities available, and time-of-day also affects travel cost, mode only affects travel cost and not opportunities. The modes cannot simply be summed, otherwise introducing a new mode would increase accessibility, even if it did not improve service. (See Red Bus, Blue Bus Paradox).

We could weight modes by shares, employing $X_{ij,m}$ which equalled the share of a given mode in a given market.

[4] (Owen and Levinson, 2015; Owen and Jiang, 2015).

	Transit Access	Auto Access	Transit Share	Multimodal Access
Before	10,000	100,000	9.1%	91,818
After	20,000	100,000	16.7%	86,667

Table 2: Illustration of Multimodal Accessibility Paradox

$$\sum_{m=1}^{M} X_{ij,m} = 1 \tag{17}$$

$$A_{i,.,.,.,.,c,t,p} = \sum_{j=1}^{J}\sum_{z=1}^{Z}\sum_{h=1}^{H}\sum_{m=1}^{M} O_{j,z,h} W_z U_h X_{ij,m} f(C_{ij,h,m,c}) \tag{18}$$

The risk with Equation 18 is that the modeshare for a slower mode might rise, lowering the value of accessibility. In the example in Table 2, there are two modes, and the transit improvement, which doesn't affect auto accessibility, by attracting additional users would reduce multimodal accessibility from 91,818 to 86,667. This is a version of Simpson's Paradox.

The Utility model, where the logsum of the utility expression in a travel mode choice model is considered the accessibility is another option. So the log of the sum of the utilities represents the value of the modes together. If a new mode were introduced that did not improve utility it would not be considered. This still has issues depending on formulation, and is impossible to measure directly, but can only be computed with a model.

Full cost accessibility (subsection 8.12) offers another way through. Once we consider the full cost of travel, ($\mathbf{C_{ij,h,m}}$) considering both short run and long run private and social costs, modes other than the automobile begin to be competitive. Accessibility here ($A^*_{i,.,.,.,.,c,t,p}$) is simply the accessibility of the best mode in a particular market.

$$A^*_{i,.,.,.,.,c,t,p} = \max_m A_{i,.,.,.,m,c,t,p} \tag{19}$$

This assumes all modes are available. People without automobiles have a reduced choice set compared to those with automobiles. People who cannot ride bikes similarly have fewer options. At the individual level, this is straight-forward to handle. Otherwise, we need to develop a matrix of modal availability ($\mathbf{V_{i,m}}$) to weight this appropriately. The value for each i, m element takes a value between 0 and 1 indicating the share of people who have access to mode m for origin i.

$$\mathbf{V_{i,m}} \in (0, 1) \tag{20}$$

This availability matrix is employed in a variation of the availability weighted mode-optimal accessibility computation ($A^*_{i,.,.,.,.,c,t,p}$):

	Transit Access	Auto Access	Transit Share	Multimodal Access
Before	10,000	100,000	9.1%	91,818
After	20,000	100,000	16.7%	86,667

Table 3: Illustration of Multimodal Accessibility Paradox

$$A^{**}_{i,.,.,.,c,t,p} = \max_m \sum_{j=1}^{J} \sum_{z=1}^{Z} \sum_{h=1}^{H} O_{j,z,h} W_z U_h \mathbf{V_{i,m}} f(\mathbf{C_{ij,h,m,c}}) \tag{21}$$

8.11 Multiple Time Slices, Multiple Activities, Multiple Modes, Multiple Groups

We have multiple subgroups, the overall for a given zone is

$$A^{**}_{i,.,.,.,c,t,.} = \max_m \frac{\sum_{j=1}^{J} \sum_{z=1}^{Z} \sum_{h=1}^{H} \sum_{p=1}^{P} O_{j,z,h} W_z U_h V_{i,m} S_{i,p} f(\mathbf{C_{ij,h,m,c}})}{\sum_{p=1}^{P} S_{i,p}} \tag{22}$$

That of itself is not especially interesting. However, if the subgroups vary in proportion by area (groups cluster spatially), then the person-weighted average for each subgroup will vary.

8.12 Full Cost

Thus far we have abstracted cost (C_{ij}). In most applications cost has been taken as individual travel time, so the primal accessibility asks, for instance, how many jobs a traveler can reach in 30 minutes of travel. While this is useful for many applications, it neglects many other costs of transport. From the user perspective, costs include monetary expenditures on travel, for instance tolls, transit fares, parking, fuel, costs of vehicle ownership, and so on. The cost of travel can be monetized (by converting time to money) or temporalized (by translating money costs to time), for instance by considering the amount of time required to work to earn enough to pay transit fares.

But from society's perspective, the aim is not to minimize user cost but society's full cost. If accessibility is to be used in evaluation, it must consider these factors. In this case, we need to consider congestion imposed on others, pollution emitted from the vehicle, danger from crash risk, noise, and infrastructure and other subsidies provided to travelers.

$$C_{ij,h,m,.} = \sum_{c=1}^{C} C_{ij,h,m,c} \tag{23}$$

Thus we can think about the full internal and full external and combined costs of travel.

Our hypothesis is that while automobile is often faster than other modes (and so has the highest time-based accessibility measure), it is unlikely to have the lowest full cost of travel.

8.13 Full Access

$$\mathbf{A_i} = \sum_{j=1}^{J} \mathbf{O_j} f(\mathbf{C_{ij}}) \tag{24}$$

In one sense, all we have done is repeated Equation 2, just making everything **bold**. But what is implied by that is that we are dealing with the matrix of weighted opportunities ($\mathbf{O_j}$) considering the different activity types (z) by times-of-day (h) and the matrix of full costs ($\mathbf{C_{ij}}$) considering modes (m) and various cost components (c).

References

Ennio Cascetta. *Transportation systems analysis: models and applications*, volume 29. Springer Science & Business Media, 2009.

Greater Sydney Commission. *A Metropolis of Three Cities*. Greater Sydney Commission, 2018.

Walter G Hansen. How accessibility shapes land use. *Journal of the American Institute of planners*, 25(2):73–76, 1959.

Andrew Owen and Haibing Jiang. Temporal sampling intervals and service frequency harmonics in transit accessibility evaluation. Technical report, University of Minnesota Accessibility Observatory, 2015.

Andrew Owen and David M Levinson. Modeling the commute mode share of transit using continuous accessibility to jobs. *Transportation Research Part A: Policy and Practice*, 74:110–122, 2015.

Enrica Papa and Pierluigi Coppola. Gravity-based accessibility measures for integrated transport-land use planning (grabam). *Accessibility Instruments for Planning Practice*, pages 117–124, 2012.

Optimizing multi-level, multi-objective airport slot-scheduling decisions

Fotios A. Katsigiannis
Centre for Transport and Logistics (CENTRAL)
Lancaster University Management School, Department of Management Science
Lancaster, United Kingdom
Email: f.katsigiannis@lancaster.ac.uk

Konstantinos G. Zografos
Centre for Transport and Logistics (CENTRAL)
Lancaster University Management School, Department of Management Science
Lancaster, United Kingdom

1. Introduction

Airport slot scheduling provides the basis for allocating airport capacity at congested airports [1]. In practice, the IATA World Scheduling Guidelines (WSG) are used as a framework for airport slot scheduling [2]. WSG recognizes four distinct slot priorities which are based on historical usage rights (historic, changes to historic, new entrant and other requests). Single [3], bi-objective [4, 5] and multi-objective [6] models have been proposed for optimizing slot-scheduling decisions. However, existing multi-objective models do not capture simultaneously the interactions of the decisions between the different levels of the slot priorities and the objectives. This approach does not allow the investigation of the potential system-wide benefits resulting from the sacrifices made at the different levels of the slot hierarchy. By addressing this issue, we introduce a new model and a solution approach, which is able to capture multi-level interaction between slot priorities.

The contribution of this paper is twofold. Firstly, it proposes a new tri-objective slot-scheduling model that considers simultaneously, total displacement, maximum displacement and demand-based fairness as described in [4]. Secondly, it introduces a multi-level programming approach, which is able to capture interactions between the slot priorities. The notion of inter-level tolerance is introduced by allowing interactions among the different types of slots. By tolerating weakly dominated or even dominated solutions at the upper levels (e.g. historic), our model yields better results at the lower levels (new entrants, others), thus resulting in improved system-wide results. Such systematic compromises in the values of the objectives of the upper decision levels satisfy the properties of Stackelberg games [7].

2. Model formulation

The notation of the tri-objective airport slot-scheduling model is presented in Table 1. The proposed model [expressions (i)-(iv)] considers the IATA's WSG and produces slot allocation solutions for the whole scheduling period.

Sets	A	Set of airlines denoted by a
	$M(M_a)$	Set of request series denoted by m (of airline a)
	$M^{Arr(Dep)}$: $M^{Arr} \cup M^{Dep} = M^{Total}$	Set of arrival (departure) series
	$P \subseteq M \times M$	Set of paired requests (m_{Dep}, m_{Arr}) indexed by p
	$D\ (D_m)$	Set of days in scheduling season (that movement m is to operate) denoted by d
	C: $\{5, 15, 60\}$	Set of capacity time intervals indexed by c
	T_c: $\{1,2, \dots, n\}$	Set of time intervals per day based on scale c indexed by t,s
	K: $\{Arr, Dep, Total\}$	Set of movement types denoted by k
Parameters	t_m	Requested time for slot series m
	$v_{t,m}^{d}$	Indicates whether request m belongs to a peak time period t on day d
	$T_{max,p}, T_{min,p}$	Maximum and minimum turnaround times of paired request p
	$u_{d,s,c}^{k}$	Capacity for movements k for a period $[s, s+c]$ on day d based on time interval c
	$a_{d,m}$	Indicates whether series m is requested on day d or not
	$\rho_\alpha = (\sum_{m \in M_\alpha} \sum_{d \in D} v_{tm}^{d}) / (\sum_{m \in M} \sum_{d \in D} v_{tm}^{d})$	The proportion of peak requests of airline a
Decision variables and objectives	$x_{t,m}$	Indicates whether series m is allocated to time t or not
	$Z_1 = \sum_{m \in M} \sum_{t \in T} \lvert t - t_m \rvert x_{t,m}$	Total displacement
	$Z_2 = \max_{\forall m \in M} \lvert t - t_m \rvert$	Maximum displacement
	$\mu_\alpha = \frac{\frac{\sum_{m \in M_\alpha} \sum_{t \in T} \lvert t - t_m \rvert x_{t,m}}{\sum_{m \in M} \sum_{t \in T} \lvert t - t_m \rvert x_{t,m}}}{\rho_\alpha}$	Fairness index expressing the displacement that airline a experiences in relation to the proportion of the peak requests that it submits
	$Z_3 = \max_{\forall a \in A} \lvert \mu_\alpha - 1 \rvert$	Maximum deviation from absolute fairness

Table 1: *Notation*

$$\min(Z_1, Z_2, Z_3) \quad \text{(i)}$$

Subject to constraints:

$$\sum_{t \in T} x_{m,t} = 1, \forall\ m \in M \quad \text{(ii)}$$

$$\sum_{m \in M^k} \sum_{t \in [s, s+c-1]} a_{d,m} x_{t,m} \leq u_{d,s,c}^{k}\ , \forall\ k \in K, d\ \in D,\ s \in T_c \quad \text{(iii)}$$

$$t_{m_{Dep}} - t_{m_{Arr}} \leq \sum_{t \in T} x_{m_{Dep}}\, t - \sum_{t \in T} x_{m_{Arr}}\, t \leq T_{max,p}\ , \forall\ p \in P \quad \text{(iv)}$$

Expression (i) minimizes the total displacement (Z_1), maximum displacement (Z_2) and maximum deviation from the absolute value of fairness (Z_3) objectives. When the fairness index (μ_α) is less than one, then airline a is experiencing less displacement in relation to the peak requests that it has submitted. On the other hand, for values of μ_α above one, the displacement that the airline will experience is greater than the proportion of its requests at peak times. Therefore, objective function Z_3 is minimised, since we would like μ_α to take values close to one (value of absolute fairness). Constraints (ii) ensure that each of the slots will be allocated to a time interval. Moreover, constraints (iii) are rolling capacity constraints for each type of movement (arrival, departures, or total movements) meaning that the total number of movements scheduled within various time intervals (5, 15 or 60 minutes), must not exceed the capacity of the airport for this movement and interval. Constraints (iv) are turnaround time constraints which define that the time difference between two paired requests, should not be less than the initially requested difference between them ($T_{min,p}$) either larger than a specified limit ($T_{max,p}$). The

representation of turnaround times in (iv) renders the utilization of precedence constraints expressed in [3] redundant.

3. Solution approach

The proposed multi-objective solution approach, transforms the tri-objective formulation of Section 2 to a single-objective optimisation problem [expressions (ii)-(vii)] by expressing objectives Z_2 and Z_3 in the form of linear constraints.

$$\min Z_1 \quad \text{(v)}$$

Subject to constraints (ii) – (iv) and:

$$\pm(t - t_m)x_{t,m} \leq \varepsilon_{Z_2}, \forall\, t \in T, m \in M \quad \text{(vi)}$$

$$\pm\left[\left(\sum_{m\in M_\alpha}\sum_{t\in T}|t - t_m|\, x_{t,m}\right) - \rho_\alpha\left(\sum_{m\in M}\sum_{t\in T}|t - t_m|\, x_{t,m}\right)\left(1 + \varepsilon_{Z_3}\right)\right] \leq 0, \forall\, a\ \in A \quad \text{(vii)}$$

Expression (v) minimizes the total displacement objective. Constraints (vi) and (vii) aid in the linearization of objectives Z_2 and Z_3 and set upper bounds to their values $(\varepsilon_{Z_2}, \varepsilon_{Z_3})$ facilitating the construction of the solution approach presented below.

For each slot priority and level of fairness, our algorithm calculates the range of efficient values of the total and maximum displacement (Z_2) objectives while using constraints (vii) to maintain the value of Z_3 below the current upper bound (ε_{Z_3}). Then, for each of the efficient maximum displacement values, it minimizes schedule displacement by maintaining objectives Z_2 and Z_3 under the current upper bounds $(\varepsilon_{Z_2}, \varepsilon_{Z_3})$ using constraints (vi) and (vii).

The described approach borrows from the concept of the Quadrant Shrinking Method (QSM) of [8], which is based on a two-dimensional, non-dominated point search (2D-NDP-Search). By applying the principle of inter-level tolerance, we solve only the first stage of the 2D-NDP-Search and filter out dominated solutions by considering schedule-wide rather than level-based Pareto optimality. To reduce computational times, we calculate the efficient bounds of maximum displacement without fairness considerations. Then, for each level of fairness, we check if the current maximum displacement value is attainable. Moreover, we impose a uniform fairness threshold among all priority levels and ensure that all requests are treated in a non-discriminatory manner.

4. Concluding remarks

To facilitate comparisons with existing solution approaches, we solve our model with and without multi-level considerations. Preliminary results suggest that the proposed solution approach generates a richer Pareto frontier of greater cardinality (12%), which gives a wider spectrum of better quality, system-wide solutions. In general, our findings suggest that by accepting systematic sacrifices at the schedules of the upper levels of the slot hierarchy, we get improved airport slot schedules and system-wide efficiency. Specifically, at the expense of imperceptible deterioration for the fairness objective, we obtain airport slot schedules with lower values for both total and maximum displacement objectives. Our analysis highlights strong trade-offs among the objectives of the proposed model, which are demonstrated with the use of appropriate graphs.

Acknowledgements

The work reported in this paper has been supported by UK's Engineering and Physical Sciences Research Council (EPSRC) and Lancaster University Management School through the Programme Grant EP/MO20258/1 "*Mathematical models and algorithms for allocating scarce airport resources (OR MASTER)*".

References

[1] Zografos, K.G., Madas, M.A., Androutsopoulos, K.N., "Increasing airport capacity utilisation through optimum slot scheduling: review of current developments and identification of future needs" in *Journal of Scheduling 20*, 3–24, 2017. https://doi.org/10.1007/s10951-016-0496-7.

[2] International Air Transport Association (IATA), "Worldwide slot guidelines. 8.1 Edition", 2018, URL https://www.iata.org/policy/slots/Documents/WSG%208.1%20-%20final.pdf.

[3] Zografos, K.G., Salouras, Y., Madas, M.A., "Dealing with the efficient allocation of scarce resources at congested airports", in *Transportation Research Part C: Emerging Technologies 21*, 244–256, 2012. https://doi.org/10.1016/j.trc.2011.10.008.

[4] Fairbrother, J., Zografos, K., "On the Development of a Fair and Efficient Slot Scheduling Mechanism at Congested Airports", TRB 2018 - TRB Annual Meeting, Washington, D.C, United States, 2018. https://doi.org/18-05366.

[5] Zografos, K.G., Androutsopoulos, K.N., Madas, M.A., "Minding the gap: Optimizing airport schedule displacement and acceptability", in *Transportation Research Part A: Policy and Practice*, 2017. https://doi.org/10.1016/j.tra.2017.09.025.

[6] Ribeiro, N.A., Jacquillat, A., Antunes, A.P., Odoni, A.R., Pita, J.P., "An optimization approach for airport slot allocation under IATA guidelines", in *Transportation Research Part B: Methodological 112*, 132–156, 2018. https://doi.org/10.1016/j.trb.2018.04.005.

[7] Lu, J., Han, J., Hu, Y., Zhang, G., "Multilevel decision-making: A survey", in *Information Sciences* 346–347, 463–487, 2016. https://doi.org/10.1016/j.ins.2016.01.084.

[8] Boland, N., Charkhgard, H., Savelsbergh, M., "The Quadrant Shrinking Method: A simple and efficient algorithm for solving tri-objective integer programs", in *European Journal of Operational Research 260*, 873–885, 2017. https://doi.org/10.1016/j.ejor.2016.03.035.

Toward Development of a Link Transmission Model for Pedestrian Networks

Tanapon Lilasathapornkit*

Department of Civil and Environmental Engineering

University of New South Wales

Email: t.lilasathapornkit@student.unsw.edu.au

Wei Liu

Department of Civil and Environmental Engineering

University of New South Wales

Meead Saberi

Department of Civil and Environmental Engineering

University of New South Wales

1 Introduction

Pedestrian modelling has attracted researchers' interest from different fields such as emergency evacuation [1], surveillance systems [2], transit infrastructure design [3], panic analysis [4] and pedestrian facilities [5]. A large part of the literature has focused on capturing pedestrian dynamic behaviour to develop more realistic pedestrian models that include lane formation [6, 7], account for counter flow, leader and follower behaviour [8], and reproduce bottleneck effects using experimental data [9]. Microscopic pedestrian models consider individuality of each walking agent and give detailed trajectory of all agents. This, however, requires high computational cost. More recently, Tordeux [10] proposed a mesoscopic stochastic pedestrian model based on hexagonal lattice containing multiple pedestrians with multi-directional fundamental diagram to regulate flow. Since mesoscopic models generally have lower computational cost, it is more suitable for large-scale pedestrian network problems. On the other hand, macroscopic pedestrian models consider agents flow without any individuality using the fundamental relationship between flow and density. Example of such macroscopic approach is Link Transmission Model (LTM) has opened up a new research approach that requires less computational effort while maintaining reasonable accuracy, mostly applied to model car traffic networks [11]. The main objective of this paper is to explore application of LTM in modelling pedestrian networks. We modify the existing macroscopic LTM framework of [11] for pedestrian modelling with the ultimate goal of real-time and large-scale simulation of walking networks.

2 Pedestrian Fundamental Diagram

Similarity between pedestrian and vehicle traffic flow fundamental diagrams can be found after scaling velocity and object size [12]. Aforementioned studies point out the importance of the self-organization phenomena that occur only in pedestrian traffic such as formation of self-organized lanes [6, 7], herding behaviour [8], cooperative behaviour to survive and symmetry breaking effect during emergency [13], and avoidance behaviour [14]. Pedestrian tends to explore space freely and not confined in designated lane, unlike car traffic [10]. Due to the difference in nature of vehicle traffic and pedestrian traffic, fundamental diagram based on vehicle traffic may (and should) not be directly applied to model pedestrians. Pedestrian fundamental diagram has been developed based on empirical data for uni-direction [15] and bi-direction [16] streams. Flotterod and Lammel [17] proposed a mathematical explanation to bi-directional fundamental diagram.

3 Link Transmission Model for Pedestrians

We build upon recent efforts by Himpe et al. [18] in development of an open source Link Transmission Model [19] with a first order node model for dynamic traffic assignment (DTA). The model initially determines shortest path, and assign flows into paths using all-or-nothing assignment. After all time steps have been calculated, a gap function is then calculated to find iterative flow adjustments to new paths until model fulfils dynamic user equilibrium (UE). To apply the existing LTM framework for pedestrian networks, we apply the following modifications as an initial effort:

1. Links must be able to traverse reversibly, accounting for bi-directional walking streams.
2. A bi-directional fundamental diagram needs to be used to accommodate for counter walking flows
3. Incorporate pedestrian crossings at intersections.

The first modification would adjust the structure of the node model and how it assigns flows into links. Here, only replicating links with reverse direction may not be suitable because pedestrians from both directions need to share the same infrastructure. Also, additional travel cost to traverse between two nodes from opposite side of the road should be added to imitate crossing the road with traffic light.

3 Data

We have obtained detailed walking network data from City of Melbourne [20] which comprises of building entrances (19,251 nodes), building centroids (14,217 nodes), walking paths (94,813 links), and land-use data (14,268 polygons). See Fig 2(a). In a preliminary study, we investigate a neighbourhood near Kensington Primary school in Melbourne. Footpaths in the area need to accommodate for both directions of pedestrian flow with bi-directional fundamental diagram. Fig 2(b) shows a model set up for the study area with four origin nodes and one destination node. Implication of turning fraction adjustment during the iterative process of the modified LTM is illustrated in Fig 3. After 10 iterations, turning fractions are adjusted to utilise a walking path that was previously ignored and overall pedestrian flow

volume seems to be more uniform. Modelling counter flow with a bi-directional fundamental diagram and additional travel cost for crossing the street are currently being studied as ongoing research.

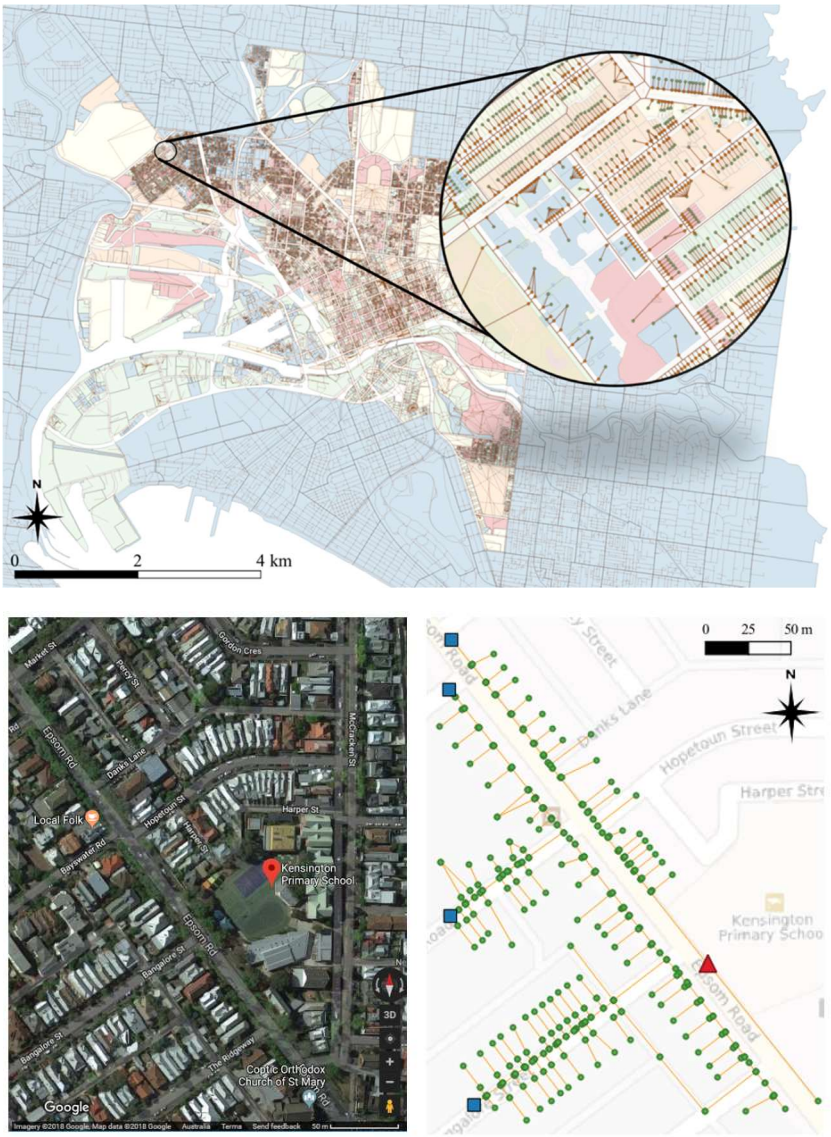


Fig. 2 (a) Top row: Melbourne Pedestrian Network Map. This map consists of 4 elements: Building entrance (orange nodes), building centroids (green nodes), land-use type (polygons), walk path (brown links); (b) Bottom row: Map of Kensington Primary school. Satellite image from Google Maps on the left showing neighbourhood around school. Model network on the right consists of nodes in green circles and links in orange lines. Blue rectangles are origin nodes and the red triangle is the destination node

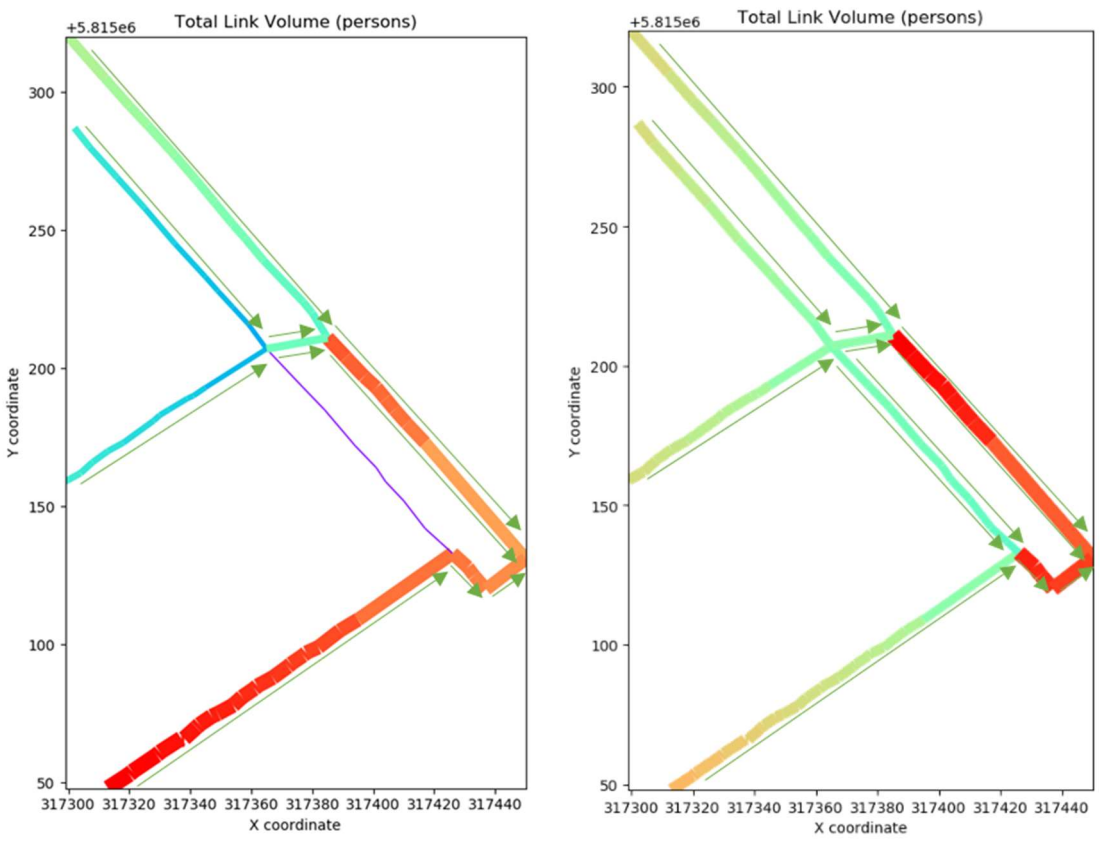


Fig. 3. Pedestrian traffic volume in each link after 1 and 10 iterations (left and right). Line thickness and colour represent volume that flow through each link where red is high volume and blue is low volume. Green arrows show direction of walking movement.

References

[1] G. Lämmel, D. Grether, and K. Nagel, "The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations," *Transportation Research Part C: Emerging Technologies,* vol. 18, no. 1, pp. 84-98, 2010.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W/sup 4/: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 809-830, 2000.

[3] S. P. Hoogendoorn and P. H. L. Bovy, "Pedestrian Travel Behavior Modeling," *Networks and Spatial Economics,* vol. 5, no. 2, pp. 193-216, 2005/06/01 2005.

[4] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature,* vol. 407, p. 487, 09/28/online 2000.

[5] F. S. Hänseler, M. Bierlaire, B. Farooq, and T. Mühlematter, "A macroscopic loading model for time-varying pedestrian flows in public walking areas," *Transportation Research Part B: Methodological,* vol. 69, pp. 60-80, 2014.

[6] M. Saberi, K. Aghabayk, and A. Sobhani, "Spatial fluctuations of pedestrian velocities in bidirectional streams: Exploring the effects of self-organization," *Physica A: Statistical Mechanics and its Applications,* vol. 434, pp. 120-128, 2015.

[7] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E,* vol. 51, no. 5, pp. 4282-4286, 1995.

[8] L. Lu, C.-Y. Chan, J. Wang, and W. Wang, "A study of pedestrian group behaviors in crowd evacuation based on an extended floor field cellular automaton model,"

Transportation Research Part C: Emerging Technologies, vol. 81, pp. 317-329, 2017/08/01/ 2017.

[9] A. Seyfried, O. Passon, B. Steffen, M. Boltes, T. Rupprecht, and W. Klingsch, "New Insights into Pedestrian Flow Through Bottlenecks," *Transportation Science,* vol. 43, no. 3, pp. 395-406, 2009.

[10] A. Tordeux, G. Lämmel, F. S. Hänseler, and B. Steffen, "A mesoscopic model for large-scale simulation of pedestrian dynamics," *Transportation Research Part C: Emerging Technologies,* vol. 93, pp. 128-147, 2018.

[11] I. Yperman, S. Logghe, and B. Immers, *The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks*. 2005.

[12] J. Zhang *et al.*, "Comparative Analysis of Pedestrian, Bicycle and Car Traffic Moving in Circuits," *Procedia - Social and Behavioral Sciences,* vol. 104, pp. 1130-1138, 2013/12/02/ 2013.

[13] K. Huang, X. Zheng, Y. Cheng, and Y. Yang, "Behavior-based cellular automaton model for pedestrian dynamics," *Applied Mathematics and Computation,* vol. 292, pp. 417-424, 2017.

[14] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and its Applications,* vol. 295, no. 3, pp. 507-525, 2001/06/15/ 2001.

[15] A. Seyfried, B. Steffen, W. Klingsch, and M. Boltes, "The fundamental diagram of pedestrian movement revisited," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2005, no. 10, pp. P10002-P10002, 2005.

[16] J. Zhang, W. Klingsch, A. Schadschneider, and A. Seyfried, "Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2012, no. 02, 2012.

[17] G. Flötteröd and G. Lämmel, "Bidirectional pedestrian fundamental diagram," *Transportation Research Part B: Methodological,* vol. 71, pp. 194-212, 2015.

[18] W. Himpe, "Integrated Algorithms for Repeated Dynamic Traffic Assignments The Iterative Link Transmission Model with Equilibrium Assignment Procedure," Geïntegreerde algoritmes voor herhaalde dynamische verkeerstoedelingen Het iterative link transmissie model met equilibrium toedelingsprocedure, Katholieke Universiteit Leuven, 2016.

[19] W. Himpe, R. Corthout, and M. J. C. Tampère, "An efficient iterative link transmission model," *Transportation Research Part B: Methodological,* vol. 92, pp. 170-190, 2016/10/01/ 2016.

[20] S. E. Planning, "City of Melbourne pedestrian network model," C. o. Melbourne, Ed., ed: City of Melbourne, 2015.

Estimating Travellers' Trip Purposes using Public Transport Data and Land Use Information

Bo Du

SMART Infrastructure Facility
University of Wollongong, Wollongong, NSW, 2522, Australia

Email: bdu@uow.edu.au

1 Introduction

In public transport system, the equipped automated fare collection (AFC) system records travellers' spatial and temporal information and generates a mass of data daily with more than ever attraction of interest and attention from both academics and practitioners. Advances in data availability and data mining techniques provide great opportunity to investigate various researches in an efficient and effective manner. A comprehensive literature review on the application of public transport smart card data before 2011 can be referred to [1]. As some relevant studies in recent years, [2] proposed a data fusion method to infer passengers' behavioral attributes of the trips based on the naive Bayes classifier model. The proposed method was applied to a single railway station in Osaka, with boarding/alighting information recorded by smart card and validation using trip survey data. [3] applied a unsupervised machine learning method, continuous hidden Markov model, to imputing the missing activities for each trip chain with integration of both clustering and transition models. [4] conducted a comparison on OD matrices between survey data and smart card data, and showed that both trip demands showed high correlation, which implied that the latter might provide a more efficient while less expensive way to construct the OD matrices.

As is well known, traditional survey serves as the major method to gather useful trip information for a long time, but it often takes high expense of manpower, time and monetary resources. Moreover, the gap between real trips and survey results can never be ignored. This study aims to investigate various travel purposes of the public transit passengers and develop a data analysis framework to estimate the trip purposes, which can be considered as an alternative or a complementarity to the traditional survey method.

2 Data Description

Singapore has a population of 5,399,000 and the major public transport forms consist of mass rapid transit (MRT), light rail transit and bus. Since the ez-link card launching in 2008, it remains the number one choice to pay transprot fare. To better illustrate passengers' travel purposes and departure features, the timeline of a single day is divided into six ranges according to the average daily ridership using public transport: early morning [4:00-7:00], morning peak [7:00-9:00], inter peak [9:00-17:00], evening peak [17:00-20:45], early night [20:45-23:00] and late night [23:00-4:00(+1day)].

To estimate trip purpose, the land use information of catchment areas of MRT stations plays an important role. In this study, five aggregated land use types are selected: commercial, residential,

business, education and others. In this study, commercial type represents locations open for customers like shopping mall and cinema; while business type represents the workplace, office, industrial factory, and so on. The proportion of each land use type at the catchment areas (circular coverage with station as center, 500m as radius) of the station is estimated based on "Singapore Master Plan 2014". With the proportion estimation of various land use features, they can be applied to replacing the alighting stations to reflect the characteristics of trip purpose. The illustration of the replacement procedure is shown in Table 1 as follows.

Table 1. Replacement of alighting station with its surrounding land use features

Trip Date	Borarding Time	Alighting Time	Alighting Station				
2013-08-12	7:40:42	7:57:27	HarbourFront				
			↓				
Trip Date	Borarding Time	Alighting Time	Commercial	Residential	Business	Education	Others
2013-08-12	7:40:42	7:57:27	65%	30%	5%	0%	0%

For illustration purpose, two weeks' smart card data of MRT North-East line (NEL) is adopted in this study. To avoid fluctuation, only data between Monday and Thursday is extracted, thus eight working days' data is adopted for analysis. Besides the five land use attributes illustrated in Table 1, the other three temporal attributes derived from smart card data include: average duration between trips, first trip start time range and last trip start time range. These five attributes are used to derive passengers' trip purposes based on a clustering method, which is introduced in the subsequent section.

3 K-prototypes Algorithm

The K-means algorithm is well known for its efficiency and simplicity, however, working only on numeric values prohibits it from being used to cluster real world data containing categorical. Since our sample data has six numeric attributes (average duration between trips, commercial, residential, business, education and others) and two categorical attributes (first trip start time range and last trip start time range), the K-prototypes algorithm is employed to handle data with mixed numeric and categorical characteristics. More details regarding the formulation can be referred to [5]. The procedure of K-prototypes algorithm is illustrated as follows:

[Step 1] Pre-given or randomly choose centroids;

[Step 2] Put each data point to its nearest centroid as a cluster based on the mixed measurement;

[Step 3] Re-calculate the centroid of each cluster based on its current data points;

[Step 4] Repeat 2 and 3 until the centroids no longer move or the iteration limitation is reached.

4 Experimental Results

With 16 stations spanning 20km, the NEL in Singapore plays an important role in weaving through the heart of the city, HarbourFront and heritage areas, such as Chinatown and Clarke Quay, through to the residential estates like Sengkang and Punggol. It is a typical MRT line with the coverage of miscellaneous trip purposes, like education, residential, work, entertainment and tourism. The goal of this study is to generate clusters with similar trip purposes. The clustering results in Table 2 shows that no extreme clusters exist, and most of the clusters are in similar size except Cluster 1's size is

relatively bigger. The columns of the result table refer to the clusters, and the first three rows indicate temporal features while the next five rows represent land use features.

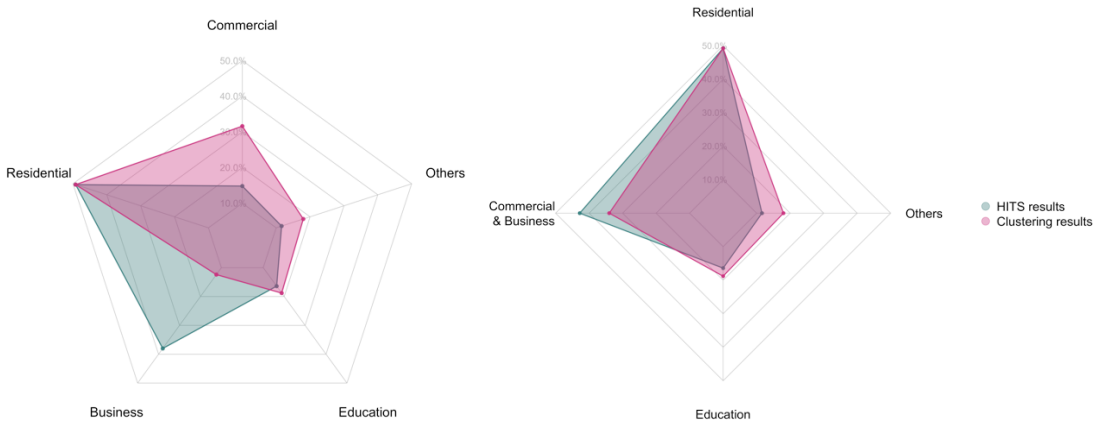
Table 2. Clustering results of trip purposes

	Cluster 1 (N= 686)	Cluster 2 (N= 299)	Cluster 3 (N= 320)	Cluster 4 (N= 339)	Cluster 5 (N= 356)
First trip start time range (average start time)	Early morning (5:30am)	Morning peak (8:00am)	Inter peak (1:00pm)	Inter Peak (1:00pm)	Inter peak (1:00pm)
Last trip start time range (average start time)	Evening peak (6:52pm)	Evening peak (6:52pm)	Inter peak (1:00pm)	Evening peak (6:52pm)	Early night (9:52pm)
Average duration (hr)	10.7	10.4	3.2	7.9	11.2
Commercial	**32.5%**	10%	**30%**	**34.7%**	**24.5%**
Residential	**47.5%**	**52.5%**	**46.3%**	**45%**	**52.5%**
Business	0%	**17.5%**	0%	0%	0%
Education	10%	**12.5%**	**11.2%**	10%	10%
Others	10%	7.5%	**12.5%**	10%	**12.5%**
Major purposes	Commercial, Residential	Residential, Business, Education	Commercial, Residential, Education, Others	Commercial, Residential	Commercial, Residential, Others

Cluster 1 shows the temporal features with first trip starting at early morning (average start time at 5:30am), last trip staring at evening peak (average start time at 6:52pm) and average duration of 10.7hr between trips, as well as land use features of residential and commercial mainly. Therefore we can infer that passengers in Cluster 1 usually travel between their residential locations and commercial areas. Similarly, Cluster 2 represents trip purposes mainly on residential, business and education. The trips of Cluster 3 are all within inter peak with short duration between trips, which indicates that the travelers in this cluster might be flexible travelers with mixed-type trip purposes and flexible schedules rather than regular commuters, and they often travel within short distance and short duration between trips. Cluster 4 represents similar trip purposes as Cluster 1, however the first trip usually happens during inter peak with shorter duration between first and last trips. Specially, tourism forms a large part of the economy (over 15 million tourists in 2014) in Singapore, therefore passengers in this cluster may include tourists. The main trip purposes in Cluster 5 include residential, commercial and others, and the trips generate later than those in the other four groups.

To validate the proposed clustering method, Household Interview Travel Survey (HITS) data with trips along NEL is used as reference. The comparison results are shown in Fig. 1. In Fig. 1(a), the HITS data was aggregated in line with the five categories in this research. However, it is worth mentioning that it is difficult to figure out the definitions of all the trip purposes in HITS data and to aggregate the trip purposes following exactly the same way in this research, hence we could notice significant difference on certain land use types, like commercial and business. In this study, commercial type represents locations open for all customers, like shopping mall. In this case, people travel to such places can either be customers or workers in those places. However, workers may belong to business type in HITS data, thus it is difficult to give a clear border between commercial and business types. In this case, commercial and business types were merged in Fig. 1(b), and we can observe similar proportions of the land use features. In general, the NEL mainly serves as a connection between residential areas and areas with business and commercial activities, which include the most famous sightseeing places in Singapore, like Sentosa, Chinatown, and so on. Similarly, many schools can be identified along this line, which explains a certain proportion

of education. All other trip purposes have been included in others type, which can be different from the definition in HITS data, as shown in the figure.



(a) Comparison based on five category (b) Comparison based on four category

Fig.1. Comparison between HITS results and clustering results

5 Conclusions

With the aid of land use information, the smart card data was analyzed to estimate passengers' trip purposes. Three temporal attributes (average duration between trips, first trip start time range and last trip start time range) and five land use attributes (commercial, residential, business, education and others) were adopted. A K-prototypes algorithm for mixed-type data was applied to obtaining the clustering results. With a MRT line in Singapore as case study, five clusters were identified to represent heterogeneous trip patterns and purposes. The proposed data analysis framework is expected to be regarded as a useful tool to impute passengers' purposes, as an alternative or a complementarity to the traditional survey approach. As a future work, numerical experiments on a large-scale public transport network will be conducted, and land use features will be adjusted to keep in line with various trip purposes in HITS data for more comprehensive and reasonable comparison.

Acknowledgements

The author thanks Land Transportation Authority of Singapore for providing relevant data support.

References

[1] M.P. Pelletier, M. Trépanier, C. Morency, "Smart card data use in public transit: A literature review", *Transportation Research Part C: Emerging Technologies* 19(4), 557-568 (2011).

[2] T. Kusakabe and Y. Asakura, "Behavioural data mining of transit smart card data: A data fusion approach". *Transportation Research Part C: Emerging Technologies* 46, 179-191 (2014).

[3] G. Han and K. Sohn, "Activity imputation for trip-chains elicited from smart-card data using a continuous hidden Markov model". *Transportation Research Part B: Methodological* 83, 121-135 (2016).

[4] C. Pineda, D. Schwarz, E. Godoy, "Comparison of passengers' behavior and aggregate demand levels on a subway system using origin-destination surveys and smartcard data". *Research in Transportation Economics* 59, 258-267 (2016).

[5] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values". *Data Mining and Knowledge Discovery* 2(3), 283-304 (1998).

Multi-reservoir MFD-based simulation: An application to the city network of Lyon

Guilhem Mariotte, **Mahendra Paipuri**, **Ludovic Leclercq** (corresponding author)
ENTPE, IFSTTAR, Univ. Lyon
F-69518, Lyon, France
Email: ludovic.leclercq@entpe.fr

Extended Abstract

In the recent past, the Macroscopic Fundamental Diagram (MFD) proved to be an attractive alternative to describe the traffic states at the network level. Several works [1, 2, 3] employed MFD-based approaches in variety of applications like perimeter control, modeling large-scale cities, *etc.* Even though [4] verified the existence of the MFD, the stability of its shape faces certain challenges like hysteresis phenomenon, heterogeneity of the traffic in urban networks. It was shown in the work of [5] that partitioning of heterogeneous networks into homogeneous regions can be a solution to obtain a well-defined MFD.

Most of the previous works on MFD approach employ the so-called accumulation-based MFD model to predict the traffic state dynamics. The hypothesis of average trip length per region is employed in the computation of system dynamics for their proposed perimeter control [3]. Recently, [6, 7] refined the idea introduced in [8] and proposed the so-called event-based approach in the framework of trip-based MFD simulation for a single reservoir system. This approach considers that all users travel at the same speed at a given time and exit the zone once they have completed their individually assigned trip length. Even though the accumulation-based approach is relatively simple and computationally less demanding, it has few shortcomings in fast-varying conditions [6] and inclusion of individual trip lengths in this approach is not trivial. On the other hand, accounting for different trip lengths is relatively straightforward in the trip-based model. However, this approach is computationally more demanding and modeling of congestion propagation in this framework is still an ongoing research question. The trip-based approach is extended to multi-reservoir systems with multiple trip lengths in each reservoir in [9].

There have been complex formulations proposed for MFD-based simulation in the literature, but very few detailed validations on real networks. Hence, this work focuses on a thorough validation

of the MFD simulators on real networks. In particular, we aim at (i) investigating the accuracy of the MFD multi-reservoir trip-based and accumulation-based models for a real large-scale network by comparing the simulation results with real traffic data; and (ii) going a step further in defining proper calibration methods for the key parameters of MFD models. The network studied is the city of Lyon, which has the second greatest urban area of France, with more than 2 million inhabitants. The perimeter studied corresponds to the urban area inside the first ring road of Lyon (also including the city of Villeurbanne). This area is manually split into 5 reservoirs exhibiting a well-defined MFD (R_1 to R_5), as presented in Figure 1(a). This area exchanges traffic with its surroundings via mainly 4 freeways related to 4 origin/destination cities, represented by four additional reservoirs: Paris (R_6), Geneva (R_7), Grenoble (R_8) and Marseille (R_9), see Figure 1(b). The demand scenario was estimated for a typical weekday in a preliminary study with a four-step model based on household trip surveys and socio-demographic data. The traffic data consists of GPS trajectories of taxi fleets in Lyon, and all the loop detectors available in the area. The data was recorded over several days, three of them were selected for this study. The large amount of taxi trips allows to determine the mean speed of each reservoir with an aggregation period of 18 min. The loop data provides a measure of the mean flow of each equipped link with the same aggregation period, which is then scaled up to each reservoir level by assuming that the measured mean flow is also representative of the non-observed links (homogeneity assumption). The average distances traveled are estimated by using both shortest path calculations and the taxi trajectories.

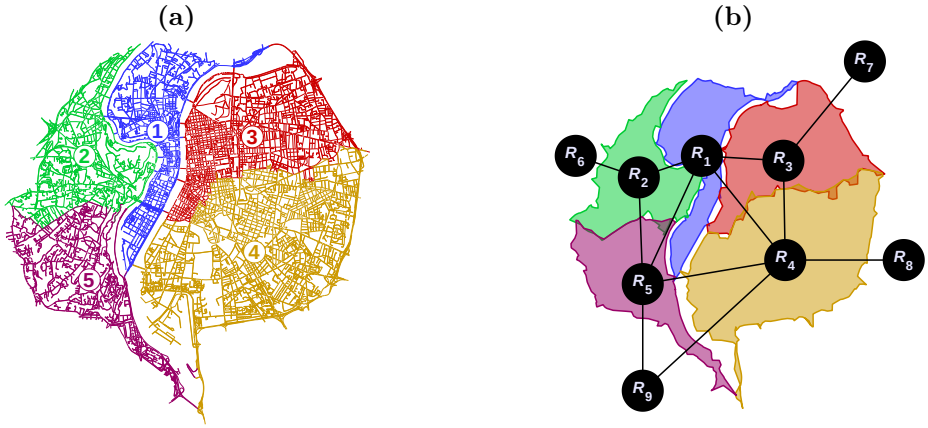


Figure 1: Network studied. (a) Road network of Lyon-Villeurbanne clustered in 5 reservoirs and (b) the reservoir configuration with external origins/destinations

A first comparison between accumulation-based MFD simulation and real data is presented in Figure 2, where the evolution of accumulation is plotted for each reservoir. While providing a reliable estimation of the accumulation level in reservoir R_1, the MFD simulation under-estimates the number of circulating vehicles in other reservoirs R_2, R_3 and R_4, with a relative error of 20-30%, see Figures 2(a)-(d). In reservoir R_1 the morning peak is better reproduced, compared to the

evening peak. Because of the scatter in reservoir R_5 between the different day datasets, it is hard to evaluate the accuracy of the simulation results in this reservoir, see Figure 2(e). Nevertheless, the overall accumulation trend and mean value are consistent with the data.

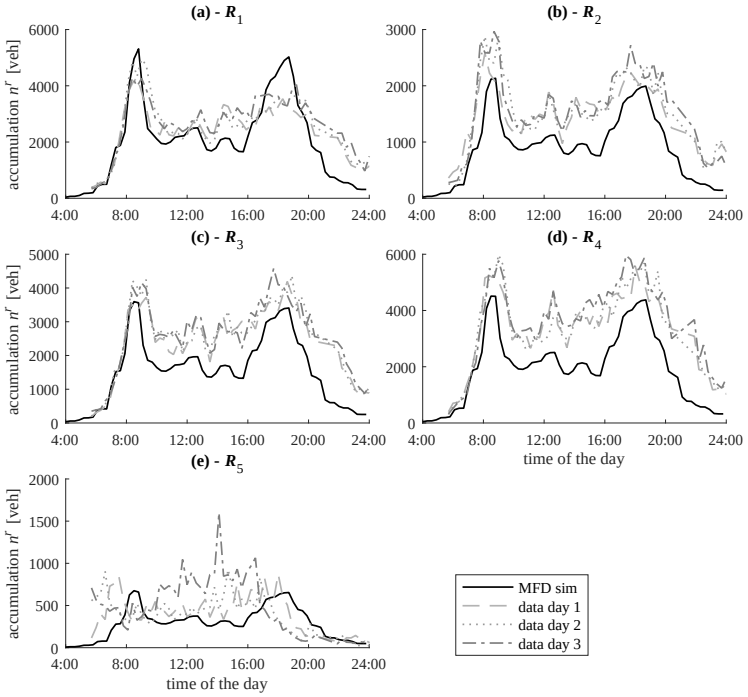


Figure 2: Comparison between real data and MFD 5-reservoir simulation. (a) Evolution of accumulation in reservoir R_1, (b) R_2, (c) R_3, (d) R_4 and (e) R_5

In this simulation test, the causes of the discrepancies between simulation and real data are likely to be multiple. The general under-estimation of accumulation levels may be due to either an under-estimation of the trip lengths in reservoirs R_2, R_3 and R_4, or an under-estimation of the demand crossing these reservoirs. Both reasons are possible, because both the estimation of the trip lengths and the demand may contain some bias. The trip lengths come from shortest path calculations on an empty network that are then adjusted based on taxi trip data. This calculation method is not necessarily representative of the real distances traveled by all vehicles. In the demand profiles, the part of traffic corresponding to trips crossing the area has been removed from our study. This was justified because we assume that these trips are mostly located on the ring road, and are thus not captured by both sources of data (the loop data and the taxi data). Actually, a small portion of them could take the city streets, which would correspond to the fraction of the accumulation we are missing. This study is still an ongoing work and up-to-date results will be presented during the conference in June.

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 646592 – MAGnUM project).

References

[1] Victor L Knoop and Serge P Hoogendoorn. Network transmission model: a dynamic traffic model at network level. In *Transportation Research Board 93rd Annual Meeting*, 14-1104, Washington DC, 2014.

[2] Mehmet Yildirimoglu and Nikolas Geroliminis. Approximating dynamic equilibrium conditions with macroscopic fundamental diagrams. *Transportation Research Part B: Methodological*, 70:186–200, 2014.

[3] Anastasios Kouvelas, Mohammadreza Saeedmanesh, and Nikolas Geroliminis. Enhancing model-based feedback perimeter control with data-driven online adaptive optimization. *Transportation Research Part B: Methodological*, 96:26–45, 2017.

[4] Nikolas Geroliminis and Carlos F. Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, November 2008.

[5] Yuxuan Ji and Nikolas Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012.

[6] Guilhem Mariotte, Ludovic Leclercq, and Jorge A. Laval. Macroscopic urban dynamics: Analytical and numerical comparisons of existing models. *Transportation Research Part B: Methodological*, 101:245–267, 2017.

[7] Raphael Lamotte and Nikolas Geroliminis. The morning commute in urban areas with heterogeneous trip lengths. *Transportation Research Procedia*, 23:591–611, 2017.

[8] Richard Arnott. A bathtub model of downtown traffic congestion. *Journal of Urban Economics*, 76:110–121, 2013.

[9] Guilhem Mariotte and Ludovic Leclercq. Mfd-based simulation: Spillbacks in multi-reservoir networks. In *Transportation Research Board 97th Annual Meeting*, 18-04679, Washington DC, 2018.

The Full Cost of Auto Accessibility

Mengying Cui
School of Civil Engineering
The University of Sydney
Email: mengying.cui@sydney.edu.au

David Levinson
School of Civil Engineering
The University of Sydney

1 Introduction

Accessibility, measuring the ability to reach valued destinations, is a reliable tool to evaluate the performance of transport systems. However, traditional accessibility measures use travel time to represent the cost of travel, which neglects the rest of the internal travel cost factors as well as the external costs of urban travel.

Cui and Levinson [1] developed a full cost accessibility (FCA) framework, which provides a theoretical basis to fill up the gap, that incorporates both internal and external costs of time, safety, emission, and money into accessibility analysis. It has the potential to change the rankings of transport investments and land developments, compared to the time-based (or time-and-money-based) accessibility evaluations, by incorporating additional cost factors, especially the cost of externalities. Many projects may be beneficial for individual travelers but present society with the expense of greater externalities.

The FCA framework has been implemented in a toy network built by Cui and Levinson [1] as a proof-of-concept. This paper, focusing on auto mobiles, extends and applies the FCA framework to the Minneapolis - St. Paul (Twin Cities) metropolitan area, which aims to, first, further demonstrate the practicality of the FCA framework for real-world applications, and, second, identify the differences and correlations between the full cost and the time-based accessibility.

2 FCA Framework

The FCA framework comprises three stages: analyzing the component costs of travel, evaluating new path types, and measuring FCA, shown in Figure 1.

The cost analysis, at first, aims to estimates the internal and external costs for each cost component, and combines them into total internal, external and full cost of travel. Cares need to be taken for the cost transfers during the combination to avoid the double counting problem.

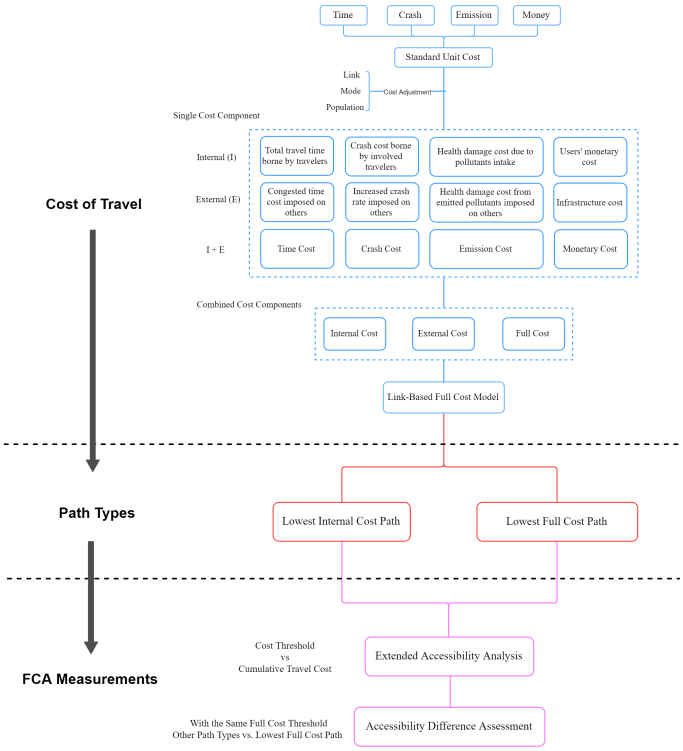


Figure 1: Full Cost Accessibility (FCA) Framework

Note that each solid blue box in the dashed box of single cost component defines the corresponding internal or external cost factor.

The lowest internal cost path and the lowest full cost path are then proposed as the optimal routes with the minimum combined internal and full costs. The cumulative internal and full costs along the lowest internal cost path and the lowest full cost path are the inputs for the accessibility calculations.

Cumulative opportunity measure, which counts the number of reachable opportunities within a given threshold [3, 4], is used for FCA measurements, written as,

$$A_{i,c} = \sum_j O_j f(C_{ij,c}) \tag{1}$$

$$f(C_{ij,c}) = \begin{cases} 1 & \text{if } C_{ij,c} \leq T_c \\ 0 & \text{if } C_{ij,c} > T_c \end{cases} \tag{2}$$

Where:

$A_{i,c}$ stands for the accessibility of origin i for cost category c ;

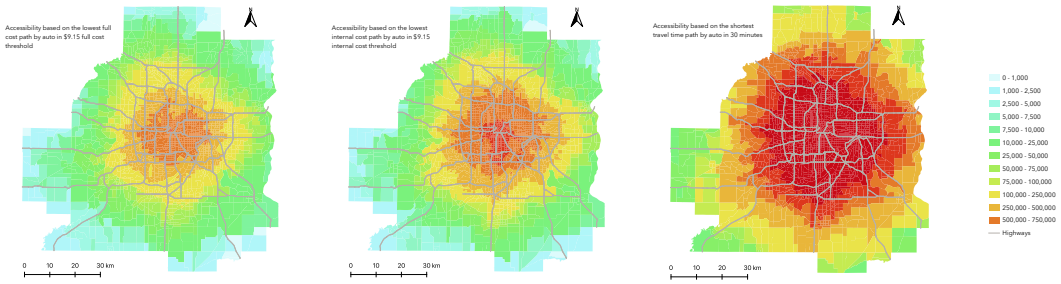
O_j stands for the number of opportunities at destination j;

$C_{ij,c}$ stands for the costs between origin i and destination j;

T_c stands for the corresponding cost threshold for cost component c.

3 Internal and Full Cost Accessibility

Cost estimates for each cost component as well as the total internal and full cost have been conducted by Cui and Levinson [2] for the Twin Cities metro area. The data are displayed in a shapefile on the basis of the TomTom road network, giving the travel cost of each link segment for all single cost components and the combined internal and full costs. Using this data, we measured the internal and full cost accessibility to jobs, see Figure 2.



(a) Full Cost Accessibility (b) Internal Cost Accessibility (c) Time-based Accessibility

Figure 2: Accessibility Measurements Based on Different Path Types in a Same Value ($9.15) of Cost Threshold

The internal and full cost accessibility show the same spatial distribution patterns as the traditional time-based accessibility that job accessibility is higher in the downtown area and decreases gradually with the increase of distance to the downtown. Comparing these three accessibility matrices, it has a clear order that time-based accessibility > internal cost accessibility > full cost accessibility with the same values of cost thresholds [1] since time cost < internal cost < full cost.

Figure 3 summarizes the correlations among the three accessibility matrices in different cost thresholds. Obviously, internal cost and full cost accessibility are highly correlated, while time-based accessibility show lower correlations with the other two, which is mainly because the time-based accessibility neglects the other 40% of the internal cost.

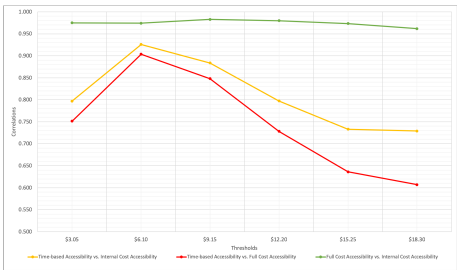
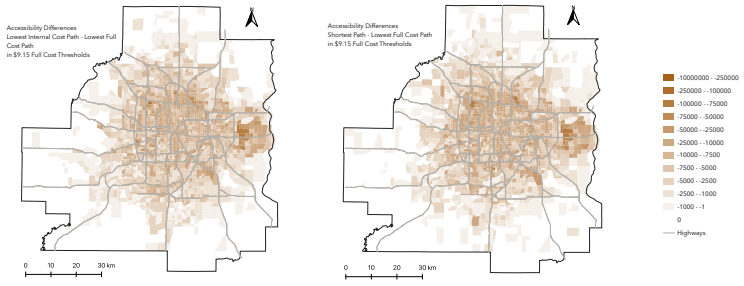


Figure 3: Correlations among Time-based, Internal Cost, and Full Cost Accessibility

[1] The value of time used in the full cost analysis is $18.3/hr

Figure 4 measures the changes of job accessibility by using the other path types rather than the lowest full cost path in the full cost thresholds. The changes are all negative since the lowest full cost path are the optimal solution with the restrictions of the full cost and using other types of path costs more from the aspect of the full cost. This implies the penalties of a travel time based route choice in terms of accessibility reductions.



(a) Lowest Internal Cost Path (b) Shortest Travel Time Path

Figure 4: Accessibility Changes in Full Cost Threshold: Other Path Types - Lowest Full Cost Path

4 Conclusion

This paper measures the full cost accessibility by auto for the Minneapolis - St. Paul metropolitan area following the steps of the full cost accessibility (FCA) framework proposed by Cui and Levinson [1]. On the basis of the previous research, this study further demonstrates the practicality of the FCA framework on real networks and identifies the correlations between the traditional time-based accessibility and the full cost accessibility. Future studies should extend the framework to other traffic modes, e.g. transit, bicycle.

References

[1] M. Cui and D. Levinson. Full cost accessibility. *Journal of Transport and Land Use*, 11:661–679, 2018.

[2] M. Cui and D. Levinson. Link-based full cost analysis of travel. *2019 TRB Annual Meeting*, 2018.

[3] R. W. Vickerman. Accessibility, attraction, and potential: A review of some concepts and their use in determining mobility. *Environment and Planning A*, 6(6):675–691, 1974.

[4] M. Wachs and T. G. Kumagai. Physical accessibility as a social indicator. *Socio-Economic Planning Sciences*, 7(5):437–456, 1973.

Optimizing Omni-Channel Fulfilment with Store Transfers

Joydeep Paul, Niels Agatz
Department of Technology and Operations Management,
Rotterdam School of Management, Erasmus University, The Netherlands

Martin Savelsbergh
Department of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, USA

Corresponding author: paul@rsm.nl

1 Introduction

Retail supply chains are changing rapidly due to the growth of e-commerce. Although online sales continue to grow, it also becomes increasingly clear that online stores will not replace the traditional brick-and-mortar stores [1]. Over the past few years, several major online retailers have extended their physical footprint, as in the case of Amazon's purchase of Whole Foods and the roll out of Amazon Go stores [2]. Thus, retailers are pursuing an omni-channel model, combining store and online channels to enhance service.

Omni-channel retail gives rise to different operational challenges. The physical stores will play a significant role in the omni-channel retail ecosystem as they form the connecting link between the online and offline channels. An increasingly popular omni-channel fulfillment model is one in which customers can pick up goods ordered online at an in-store pick-up point (PUP). The PUPs are typically not supplied from the store inventory but by a dedicated e-fulfillment warehouse. This often means that the stores are visited by multiple vehicles to replenish the store inventory and to supply the PUPs. Motivated by the fulfillment challenges at the leading omni-channel grocery retailer in the Netherlands, we develop a strategy to consolidate product flows to the stores by sharing the capacity of vehicles across different channels using the stores as potential transfer points.

This works as follows. Consider an omni-channel retailer that plans one *fixed schedule* for store replenishment and a *flexible schedule* for the supply of the PUPs. The sequence of stops (i.e., the stores visited) in the fixed schedule are determined far in advance, whereas the flexible schedule is determined daily based on the actual demand. If there is capacity available on the vehicles executing the fixed schedule, the online demand of the *shared store locations* may be transferred from the vehicles executing the flexible routes to the vehicles executing the fixed routes. This can reduce the system-wide travel costs and the number of store visits.

Our research builds on the work of Paul et al. [3], which considers a simpler setting in which a transfer can only take place at the starting location of the fixed schedule (i.e., the warehouse) by using dedicated transfer trips. In this paper, we allow every store location to be a possible *transfer point*. This involves deciding the transfer points as well as synchronizing the two schedules. Allowing transfer of demand at different stores increases the potential for capacity sharing as more capacity becomes available for transfers later in the fixed schedule, after delivering to some store locations. Another advantage is that the use of transfers may result in a shorter flexible schedule.

We introduce the Shared Capacity Routing Problem with Transfers (SCRPT) in which the goal is to design a flexible schedule that delivers to all required store locations, either directly or by transferring to the fixed schedule, and that minimizes costs.

2 Problem Definition

We consider a stylized abstraction of our real-life case with a single vehicle for both the fixed and the flexible schedule. We model the SCRPT on a complete graph $G = (V, A)$. Here, $V = N \bigcup \{o\} \bigcup \{d\}$, where o and d are the warehouses of the fixed and flexible schedule respectively and N is the set of store locations visited in the *fixed route*. Let $S \subseteq N$ be the set of store locations that need to be served in the *flexible route*, where $i \in S$ has a demand $q_i \geq 0$ (which needs to be fulfilled from the warehouse d). Demand of each store $i \in S$ can be fulfilled by direct delivery or by transferring to the fixed route if feasible. The costs of traversing arcs, $c_{ij} \forall (i, j) \in A$, satisfy the triangle inequality.

The demands of stores can be transferred at a location $i \in N \bigcup \{o\}$, which we refer to as a *transfer-point*. Only stores $j \in S$ which are visited after i in the fixed route can be transferred at i. The demand of a store j can be transferred in parts at multiple stores $i < j$. For example, half of the demand of store 4 can be transferred at store 1 and another half at store 2. Let T_i be the set of the stores whose demand (full or partial) can be transferred at i. For a time-feasible transfer at location i, the time of arrival, t_i, at location $i \in N \bigcup \{o\}$ in the flexible route cannot be later than the time of arrival, a_i, at location i in the fixed route. Note that a transfer can take place at a location i that does not need to be served in the flexible route, i.e., $i \in N \bigcup \{o\}, i \notin S$ can be a possible transfer-point.

At every store location $i \in N$, more capacity becomes available in the vehicle associated with the fixed route as it drops off the demand d_i associated with the store of the fixed route, which can be used to receive transfers from the flexible route. However, there should be enough space available at the store location $i \in N$ to handle the transfers. Let e_i^v and e_i^s denote the *vehicle spare capacity* in the fixed route and *store handling capacity* at location $i \in N \bigcup \{o\}$, respectively. The net *transfer capacity*, e_i, at any location i that can be used for handling the transfers, is the minimum of these two capacities, i.e., $e_i = min(e_i^v, e_i^s)$. Note that when the store handling capacity is never limiting, then the transfer capacity e_i is always increasing along the fixed route.

The demands of stores in $T_i \subseteq S$ can be transferred at i (full or partial) as long as the total demand does not exceed the transfer capacity e_i. We refer to a set of stores whose

demand is capacity feasible to be transferred at location i as an *i-transfer*.

The cost of the fixed route is exogenous to the model, so it suffices to minimize the transport costs of the flexible route. The goal of the SCRPT is to determine the set of transfer-points with their corresponding i-transfer sets and the route sequence for the non-transferred stores and the transfer-points so that total costs are minimized.

3 Solution Approaches

We develop a mixed integer linear programming (MILP) model which can solve instances of small size. As the SCRPT with one flexible route reduces to the travelling salesman problem when there is no spare capacity in the fixed route, the SCRPT is NP-hard. Hence, we focus on developing heuristics to solve the problem for large instances.

We present optimal solution approaches for a special case of the SCRPT in which the the stores are located on a circle and the warehouses of two schedules are co-located. In that case, we can build an auxiliary graph with store locations as nodes and feasible "transfer short-cuts" as arcs. We use several dominance rules to reduce the number of arcs that can possibly be part of an optimal solution. A shortest path in the auxiliary graph gives an optimal flexible route. We use the insights from this special case to develop a heuristic for the general case when store locations need not be on a circle and the warehouses are not co-located. When the warehouses are not co-located, we first find the location(s) $l \in N \bigcup \{o\}$ where the flexible route can *catch-up* with the fixed route, then find the shortest path between l and $n = |N|$ using the auxiliary graph and finally, build a Hamiltonian path for the stores ($i \in S, 0 < i < l$[1]) that are not visited before the catch-up location. In the next section, we benchmark our heuristic against the optimal solutions and also, present the potential savings that can be obtained by capacity sharing.

4 Results & Discussion

We generate instances with store locations randomly distributed on an euclidean plane. The stores are categorized into small, medium and large, based on their capacity to handle transfers. We solve a TSP to get the fixed route visiting all store locations. We assume that the vehicle associated with the fixed route is full when it starts the route. The spare capacity in the vehicle of the fixed route becomes available when the demand from the fixed schedule is dropped off at a location. The ratio of the average demand of stores in the flexible schedule and in the fixed schedule impacts the potential benefit of exploiting transfers and capacity sharing. We use different values of the ratio to generate the instances.

In Table 1, we report the performance of the heuristic in terms of the gap from the optimal solution (obtained by solving the MILP). The average optimality gap is 2.4%, while the heuristic finds the optimal solution 24 out of the 45 instances.

In Table 2, we first show the number of stores whose demands are transferred to the fixed schedule and the associated savings in transport cost using stores as transfer points.

[1] stores are indexed in order of their visit in the fixed route

We see that the average savings in transport costs across all instances are around 61.1%.

We also compare these savings with the savings obtained using the warehouse of the fixed schedule as the single transfer point. For this experiment, we assume there is spare capacity in the vehicle of the fixed route at the starting location to accommodate 20% of the total demand of the stores on the flexible route. With the stores as potential transfer points, the savings in transport cost increase on average to 61.1% from 4.6%. When we use the stores as transfer points, the transfer options increase as can be observed in the increase in the number of stores whose demands are transferred.

Table 1: Performance of heuristic

Instance Size	Optimality Gap* (%)
30	2.0
35	0.9
40	4.4
Average	**2.4**
# of times optimal found	**24** / 45

* average of 15 replications

Table 2: Savings due to capacity sharing via transfer points

Instance Size	Multiple Transfer		Single Transfer	
	# of stores transferred	Savings (%)	# of stores transferred	Savings (%)
30	17	59.3	4	0.6
40	20	62.3	5	2.8
50	27	59.8	7	4.1
60	26	59.5	8	7.0
70	28	61.5	9	7.2
80	34	64.5	11	6.1
Average		61.1		4.6

The initial results of our experiments show potential savings due to transferring of store demands by sharing capacity across channels. We plan to improve the heuristic to further reduce the average optimality gap. We will conduct an extensive computational study to understand the effect of capacity sharing under different settings.

References

[1] M. Brown, D. Farmer, and N. Ganenthiran. "Recasting the Retail Store in Todays Omnichannel World." AT Kearney (2013).

[2] Nat Levy. 2018. https://www.geekwire.com/2018/amazons-expanding-u-s-brick-mortar-footprint-stacks-big-retailers/

[3] J. Paul, N. Agatz, R. Spliet and R. De Koster, "Shared Capacity Routing Problem - An omni-channel retail study", *European Journal of Operational Research*, 2018. https://doi.org/10.1016/j.ejor.2018.08.027.

Perimeter Flow Control with Time-Varying Cordon based on Macroscopic Fundamental Diagram

Ye Li
The University of Sydney
School of Civil Engineering

Reza Mohajerpoor
The University of Sydney
School of Civil Engineering

Mohsen Ramezani
The University of Sydney
School of Civil Engineering
Email: mohsen.ramezani@sydney.edu.au

1 Introduction

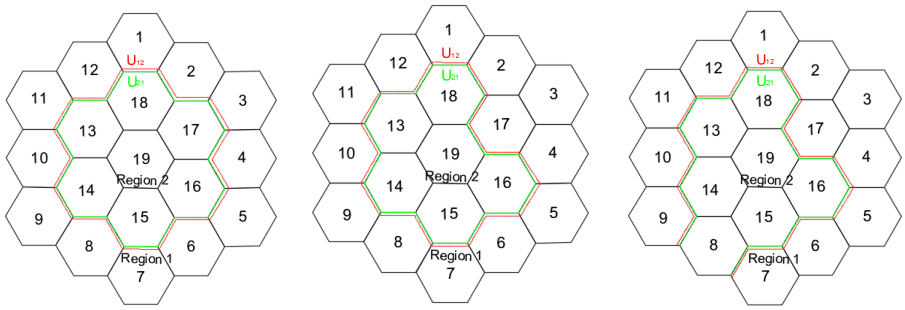
To improve traffic operation efficiency in large-scale urban networks, numerous traffic signal control policies were developed and implemented in the past few decades. However, these signal control policies are nominally designed for isolated intersection control or coordinated control in arterials. A very recent approach to extend the spatial extent of traffic signal control to the network level is perimeter flow control based on the Macroscopic Fundamental Diagram (MFD) model. MFD describes a well-defined, low-scattered, and non-linear relationship between mean weighted flow and vehicle accumulation of a network where the spatial distribution of congestion is homogeneous.

Perimeter flow control is an effective traffic control method that monitors vehicle density in a protected region (PR) and manipulates the traffic inflow to the PR to regulate vehicle accumulation under a certain value. Previous studies demonstrated that perimeter control can minimize the network total delay, e.g. [1]. The existing perimeter flow control methods are based on static, time-invariant cordon (see [2], [3]), i.e. the region boundaries are fixed, which leads to limited consideration of convoluted temporal changes in vehicle accumulation distribution. In this study, we propose a perimeter flow control with time-varying cordon to reduce the total delay in a two-region network by adjusting the PR boundaries over time. The cordon selection algorithm associates an index to each subregion quantifying the extent that the subregion needs protection from hypercongestion. Consequently, the algorithm clusters all the subregions either to the PR and a peripheral region such that the protection index of subregions inside the PR has the maximum difference from

the protection index of subregions in the peripheral region. Ultimately, a Proportional-Integral (PI) regulator is employed to acquire the optimal control values.

2 Methodology

In this study, we consider an urban network that is partitioned into a number of subregions each with a well-defined MFD, as shown in Figure 1. *Region 1* is the peripheral region and *Region 2* is the protected region. The perimeter controllers are implemented at the boundaries between the two regions that are denoted by U_{12} and U_{21}.



(a) (b) (c)

Figure 1: The perimeter control with time-invariant boundaries between the two regions, *Region 1* includes subregion 1 to 12 and *Region 2* includes subregion 13 to 19, (b) one possible new cordon between the two regions where subregion 17 is included in *Region 1* instead of *Region 2*, and (c) one possible new cordon between the two regions where subregion 8 is included in *Region 2* instead of *Region 1*.

2.1 Traffic flow model based on MFD dynamics

The traffic flow model is developed based on the MFD to describe traffic propagation dynamics among the subregions. The internal trip completion flow of subregion i at time t that does not leave subregion i is denoted by $m^i_{ii}(t)$ (veh/s) that is defined as $m^i_{ii}(t) = \theta^i_{ii}(t) \cdot n_{ii}(t)/n_i(t) \cdot p_i(n_i(t))/l_{ii}(t)$. $n_i(t)$ is the accumulation of subregion i at time t and $p_i(n_i(t))$ is the MFD production (weighted flow) of subregion i at time t. Let ϕ_i denote the set of subregions directly reachable from subregion i. Thus, the transfer flow from subregion i with destination subregion i through subregion h is denoted by $m^h_{ii}(t)$ (veh/s), where $m^h_{ii}(t) = \theta^h_{ii}(t) \cdot n_{ii}(t)/n_i(t) \cdot p_i(n_i(t))/l_{ih}(t)$; $h \in \phi_i$. Similarly the transfer flow from subregion i to the next immediate subregion h with final destination subregion j is $m^h_{ij}(t) = \theta^h_{ij}(t) \cdot n_{ij}(t)/n_i(t) \cdot p_i(n_i(t))/l_{ih}(t)$; $h \in \phi_i$; $i \neq j$. $\theta^h_{ij}(t)$ denotes the percentage of total transfer flow from subregion i with final destination subregion j through subregion h such that $h \in \phi_i$. Accordingly, $\theta^i_{ii}(t) + \theta^h_{ii}(t) = 1$ for $h \in \phi_i$, and $\sum_{h \in \phi_i} \theta^h_{ij}(t)$

$= 1$ for $\forall i, j$. $n_{ij}(t)$ denotes the accumulation in subregion i with final destination in subregion j. Evidently, $n_i(t) = \sum_{j \in R} n_{ij}(t)$ where R denotes the set of all subregions.

$l_{ij}(t)$ represents the average trip length from subregion i to subregion j. We assume the trip lengths inside subregions are constant, that is $l_{ii}(t) = l_{ih}(t) = l_i$. Furthermore, we assume $\theta^i_{ii}(t) = 1$ and $\theta^h_{ii}(t) = 0$ that is the internal trips at the final subregion do not leave the destination subregion. The proposed model accommodates the subregion receiving capacity according to [2] (similar to Cell Transmission Model), which limits the transfer flows accordingly ($m^h_{ii}(t)$ and $m^h_{ij}(t)$ to $\hat{m}^h_{ii}(t)$ and $\hat{m}^h_{ij}(t)$). The perimeter controllers are applied on the border between the two regions controlling the transfer flow between associated subregions. For instance $u_{ih}(t)$ control the transfer flow between subregion i and subregion h. Note that $0 \leq u_{\min} \leq u_{ih}(t) \leq u_{\max} \leq 1$ to reflect the physical constraints on the minimum and maximum possible values of the perimeter control. Consequently, the subregional vehicle conservation equations are as follows,

$$\frac{\mathrm{d}n_{ii}(t)}{\mathrm{d}t} = q_{ii}(t) - m^i_{ii}(t) - \sum_{h \in \phi_i} u_{ih}(t) \cdot \hat{m}^h_{ii}(t) + \sum_{h \in \phi_i} u_{hi}(t) \cdot \hat{m}^i_{hi}(t) \tag{1}$$

$$\frac{\mathrm{d}n_{ij}(t)}{\mathrm{d}t} = q_{ij}(t) - \sum_{h \in \phi_i} u_{ih}(t) \cdot \hat{m}^h_{ij}(t) + \sum_{h \in \phi_i} u_{hi}(t) \cdot \hat{m}^i_{hj}(t), \quad i \neq j. \tag{2}$$

where $q_{ij}(t)$ are demands from subregion i to subregion j.

2.2 Cordon selection algorithm

In this study, we propose a perimeter control strategy with time-varying cordon to reduce the total delay in a two-region network by adjusting the protected region boundaries every ΔT minutes. The details of the cordon selection algorithm is as follows. We define the subregion weight, w_i as

$$w_i(t) = (\frac{n_i(t)}{n^{\mathrm{cr}}_i})^2 \tag{3}$$

where n^{cr}_i is the subregion i critical accumulation based on MFD. Let $|\phi_j|$ denote the number of subregions adjacent to subregion j; $s_j(t)$ is an estimation of sending flow from subregion j towards subregion i

$$s_j(t) = \frac{p_j(n_j(t))}{p^{\max}_j \cdot (|\phi_j| + 1)}, \quad \forall j \in \phi_i. \tag{4}$$

where $p^{\max}_j$ is the maximum production of subregion j based on the MFD of subregion j. Therefore, the protection index of subregion i is defined as $c_i(t) = w_i(t) \cdot (\Sigma_{j \in \phi_i}\ s_j(t))$, $\forall i \in R$. The protection index, $c_i(t)$, considers two factors, (i) subregion accumulation, $n_i(t)$, to reflect the subregion congestion level and (ii) neighbor subregions outflow towards the subregion.

The regional protection index is determined by simply counting all related subregion protection indexes. The average protection index of *Region 1* and *Region 2* can be acquired through $\overline{C}_1(t) = \sum_{i \in R_1} c_i(t)\ /\ |R_1(t)|$ and $\overline{C}_2(t) = \sum_{i \in R_2} c_i(t)\ /\ |R_2(t)|$. The number of subregions in *Region 1* and *Region 2* at time t are $|R_1(t)|$ and $|R_2(t)|$, respectively. Note that $R_1(t) \cup R_2(t) = R$.

The algorithm clusters all the subregions into a PR and a peripheral region such that the average protection index of *Region 2* has the maximum difference from the average protection index of *Region 1*, which is represented by $DCI(t) = \max|\overline{C}_2(t) - \overline{C}_1(t)|$. We assume at most only one subregion can be switched between the regions at each time step. Subregions cannot switch to a new region that are not directly connected to the previous region boundaries (e.g. subregion 19 in original Region 2 as shown in Figure 1(a)). The regions with the maximum $DCI(t)$ is the new region configuration for the perimeter control.

3 Preliminary Results

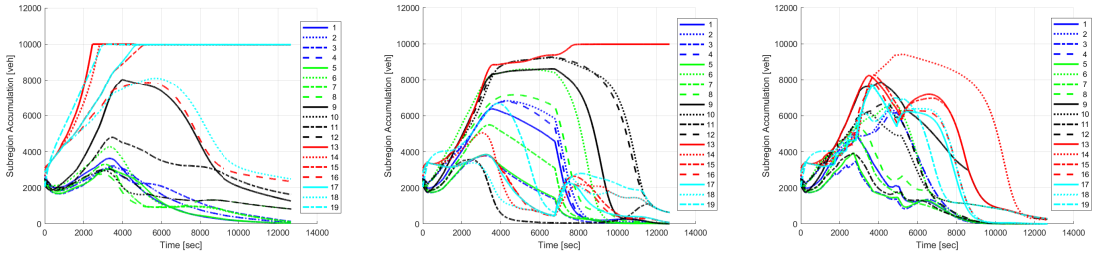
In order to highlight the effectiveness of time-varying cordon, we compare three scenarios, (i) no control, (ii) perimeter control with static cordon, and (iii) perimeter control with dynamic time-varying cordon. The two latter control approaches are based on a Proportional-Integral (PI) regulator, which is presented in (5). K_P and K_I are gain values, *e(t)* is the error between the current measurement of Region 2 accumulation and the desired value, i.e. *e(t)*=$N_2(t) - N_2^{\text{cr}}(t)$. $N_2^{\text{cr}}(t)$ is the critical accumulation of Region 2 that is time-varying because the number of subregions in this region changes as cordon changes. $U_{12}(t)$ is the perimeter control manipulating the transfer flow between Region 1 to Region 2 to maintain the accumulation of Region 2 close to its critical value, while $U_{\min} \leq U_{12}(t) \leq U_{\max}$.

$$U_{12}(t) = U_{12}(t-1) - K_P \cdot (e(t) - e(t-1)) - K_I \cdot e(t) \tag{5}$$

Results of the three scenarios pinpoint the importance of applying time-varying cordon in a multi-region traffic network. The evolution of subregion accumulations over the studied period represent that the time-varying cordon reduces the subregion accumulation heterogeneity significantly, as shown in Figure 2. The total network delay is summarized in Table 1. It is obvious that the total network delay is decreased while implementing either control strategies, while dynamic cordon perimeter control performs superior to the static one, with 14% reduction in total delay. Hence, time-varying cordon achieves the goal to distribute the accumulations more homogeneous than static cordon and improves the traffic network efficiency.

Table 1: Total network delay for different control strategies (10^6 (veh·s)). Values in parentheses show the improvement over the No Control case.

No Control	Perimeter Control with Static Cordon	Perimeter Control with Dynamic Cordon
1024.70	783.93 (22.03%)	674.01 (34.22%)



(a) (b) (c)

Figure 2: Subregion accumulations over time: (a) no control, (b) perimeter control with static cordon, and (c) perimeter control with dynamic cordon.

References

[1] N. Geroliminis and J. Haddad and M. Ramezani, "Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: a model predictive approach", *IEEE Transaction on Intelligent Transportation Systems* 14, 348-359 (2012)

[2] M. Ramezani, J. Haddad and N. Geroliminis, "Dynamics of heterogeneity in urban networks: Aggregated traffic modeling and hierarchical control", *Transportation Research Part B: Methodological* 74, 1-19 (2015)

[3] K. Aboudolas and N. Geroliminis, "Perimeter and boundary flow control in multi-reservoir heterogeneous networks", *Transportation Research Part B: Methodological* 55, 265-281 (2013)

A predictive model of lane-changing possibilities: deep learning approach

Seunghyeon Lee

Department of Civil and Natural Resources Engineering,

University of Canterbury, Christchurch, New Zealand

Dong Ngoduy

Department of Civil and Natural Resources Engineering,

University of Canterbury, Christchurch, New Zealand

Email: seunghyeon226@gmail.com

The objective of this study is to develop a deep learning algorithm for estimating a real-time possibility of lane-changing (LC) behaviour in a continuous stochastic car-following model. The proposed modelling framework aims to cope with probabilistic characteristics of lane-changing manoeuvres in a freeway. There are five distinctive contributions of this study; 1) a stochastic volatility derived from LC manoeuvres is integrated into a multi-lane stochastic car-following model, 2) the CNN (Convolutional Neural Network) is used to estimate a probability of LC manoeuvres in the integrated multi-dimensional car-following model, 3) imaged second-based trajectories of the lane-changer and surrounding vehicles are used to identify whether LC manoeuvres occurred by using the CNN, 4) the proposed method paves the way for an applicability of the integrated multi-lane car-following model for multi-lane Cooperative Adaptive Cruise Control (CACC) as well as connected traffic systems, and 5) the proposed method is validated using a real world high-resolution vehicle trajectory dataset.

Our previous study [1] and [2] provides its firm foundation of applicability of Langevin equations into a stochastic continuous car-following model, which is a key element to construct multi-dimensional interactions between vehicles on a road in this study. An integrated form of stochastic differential equations (SDEs) for the acceleration of the *n*th vehicle in the *k*th lane, including a longitudinal and a lateral interaction between vehicles on the road, is defined as below:

$$\frac{dv_{n,k}}{dt} = \frac{dv_{n,k}^{\text{Long}}}{dt} + \left[\left(\Lambda_{n,k} \frac{dv_{n,k}^{\text{Lat}}}{dt} \right) + \left(\Lambda_{n-1,k} \frac{dv_{n-1,k}^{\text{Lat}}}{dt} \right) - \left(\Lambda_{n-1,k+1} \frac{dv_{n-1,k+1}^{\text{Lat}}}{dt} \right) \right] \quad (1)$$

where $v_{n,k}^{\text{Long}}$ and $v_{n,k}^{\text{Lat}}$ are the longitudinal and lateral speed of the *n*th vehicle in the *k*th lane, respectively. Furthermore, the lateral acceleration of the leading (*n*–1)th vehicles, $dv_{n-1,k}^{\text{Lat}} / dt$ and $dv_{n-1,k-1}^{\text{Lat}} / dt$, on both the initial and the target lanes are created to reflect the preceding lane-changer's impacts on the subject vehicle on the propose framework. The lateral acceleration of the vehicles are multiplied by the corresponding probability of LC manoeuvres, $\Lambda_{n,k}$, which is used as a sensitivity variable in the SDEs. These corresponding probabilities are calculated by the data-driven deep learning

method, the CNN, using space-headway, velocity and acceleration differences between lane-changer and surrounding vehicles. The sensitivity of LC manoeuvres to the multi-dimensional stochastic car-following model is determined according to the corresponding value of LC probabilities. In addition, $v_{n,k}$ is the actual velocity of the nth vehicle in the kth lane, which depends on both longitudinal and lateral interactions between vehicles. The detailed explanations are provided in [1] and [2].

The main elements of the input layer are differences of positions, velocities, and accelerations between the lane-changer and the surrounding vehicles, including preceding vehicles and following vehicles in the initial and the target lane, for 20 frames in a single frame as 0.1s. The difference of positions between the subject vehicle and the surrounding vehicles is shown as the space-headway of the lane-changer to the surrounding vehicles. In the meantime, a gap of velocity and acceleration of the lane-changer with these trajectory quantities of vehicles in the neighbourhood is calculated as these traffic quantities of the neighbourhood subtracted from that of the lane-changer. The three kinds of traffic quantities are combined into one input layer for the CNN. Due to the flexibility of the CNN algorithm against the size of the input layer, we will utilize the varying size of the input layer, which depends on the number of traffic quantities. These include the space-headway, velocity, and acceleration in this study, the number of surrounding vehicles, which are the front and the rear vehicles, the number of time frame, and the number of considering adjacent lanes, which are the initial and the target lanes in this study, for each trajectory of the target vehicle in a time slot. Accordingly, the size of the input layer is determined as $3 \times 2 \times \mathbf{T} \times \mathbf{K}$. We define the input layer as the following equations.

$$z_{ktc} = \begin{cases} \left|x_{n,k}(t) - x_{n,k-1}(t)\right|, \left|x_{n,k}(t) - x_{n,k+1}(t)\right|, \left|x_{n,k}(t) - x_{n+1,k-1}(t)\right|, \left|x_{n,k}(t) - x_{n+1,k+1}(t)\right|, \text{for } c=1 \\ v_{n,k}(t) - v_{n,k-1}(t),\ v_{n,k}(t) - v_{n,k+1}(t),\ v_{n,k}(t) - v_{n+1,k-1}(t),\ v_{n,k}(t) - v_{n+1,k+1}(t), \text{ for } c=2 \\ a_{n,k}(t) - a_{n,k-1}(t),\ a_{n,k}(t) - a_{n,k+1}(t),\ a_{n,k}(t) - a_{n+1,k-1}(t),\ a_{n,k}(t) - a_{n+1,k+1}(t), \text{ for } c=3 \end{cases} \quad (2)$$

$$\forall t \in \mathrm{T}, \forall k \in \mathrm{K}, \forall n \in \mathrm{N}$$

$$l = 1 \Rightarrow z_{ktc} = y_{ijc} \left(\forall z_{ktc} \in \mathbf{Z}^{\mathrm{cnn},n}, \forall y_{ijc}^{l} \in \mathbf{Y}^{l} \right) \quad (3)$$

In Equation (2), each cell of imaged data set, z_{ktc}, is illustrated as a series of 3 channels, including space-headway, speed gaps, and acceleration gaps between the target vehicle and the surrounding vehicles. The value of the initial input layer is identical to the value of the inputs to the first layer illustrated in Equation (3), in which the size of one channel in the initial input layer is the number of surrounding vehicles at the target and the initial lanes times the length of the time frame in milliseconds ($2 \times \mathbf{T} \times \mathbf{K}$). The input value to the lth layer, y^{l}_{ijc}, is explained in the section of a convolutional layer in the ensuing paper. Moreover, y^{l}_{ijc}, the value of input in column i in row j in channel c in the vector $\mathbf{Y}^{l}$, to the lth layer is illustrated in the convolutional procedure as below:

$$y_{ijc}^{l} = \sum_{u} \sum_{v} w_{uv}^{l} \cdot y_{i+u,j+v,c}^{l-1} + b^{l}, \forall c \in \mathrm{C} \text{ with } y_{ijc}^{l} = f\left(y_{i+u,j+v,c}^{l-1} \right), \quad (4)$$

where w_{uv}^{l} represents a kernel matrix (i.e., weight matrix) of dimension $\mathbf{U} \times \mathbf{V}$ at the lth layer connecting neurons of the lth layer with them in the (l – 1) th layer. A bias matrix of the layer l, is defined as b^{l}. The output vector at the lth layer, y_{ijc}^{l}, is defined by $f\left(y_{i+u,j+v,c}^{l-1} \right)$, where $f(\cdot)$ represents an elementwise activation function. In general, either the logistic sigmoid function, $f(x) = \left(1 - e^{-\beta x}\right)^{-1}$, or the hyperbolic

tangent function, $f(x) = a\tanh(bx)$, or the rectified linear unit (ReLU) function, $f(x) = \max(0, x)$, can be used as the activation function, $f(\cdot)$.

To verify the effectiveness of the proposed methods, we apply a stochastic car-following model with stochastic volatility derived from LC to real traffic trajectory data set collected from U.S. 101 freeway from 7:50 a.m. to 8:35 a.m. on June 15, 2006. The dataset and detailed information were provided by the Federal Highway Administration's Next Generation Simulation (NGSIM). We select cases of LC manoeuvres without LC manoeuvres of the leading and the following vehicle of a lane changer for 100 time-frames (10s) before and after the LC manoeuvre (i.e. total 00 time-frames) in all lanes of the target section.

We construct the architecture of the CNN for a classification of scales for LC probabilities modified from LeNet-5 in [3] using Python with Keras library. Here it is emphasized that multilayer networks can be capable of learning complex, high-dimensional, nonlinear mappings from large coupled image recognition tasks with gradient descent. The trained LeNet showed the excellent performance to categorize hand-written features into designated classes, while it required less computational burdens than fully-connected single network problems. The primary elements of the LeNet are local receptive fields, shared weights, and spatial subsampling. LeNet includes two convolutional layers connected with subsampling layers and the final double fully-connected layers linked to the Gaussian-connected layer. The first layer is the 2-dimentional convolutional layer with ReLU activation function connected with the 2-dimentional max-pooling layer as the second layer. And then, we design five fully connected layers with 30% as the rate of dropout. We set a ReLU function to an activation function in four layers except the final layer with a Sigmoid function. 195 cases of LC manoeuvres from 7:50 a.m. to 8:05 a.m. are used for training the CNN model, whereas 100 cases of LC manoeuvres from 8:20 a.m. to 8:35 a.m. are used for the validation process. The total loss of the validation set derived from binary cross-entropy method in Keras is 0.345, whereas the accuracy of the validation set is 0.7959.

To compare the estimates and the observed in both train and test sets, the detailed statistics are provided in the following contingency table. We exclude the period, when the LC probability lasted as zero for a while, for all cases from the contingency table.

Table 1. A contingency table of the proposed CNN method.

			$\hat{p}_{n,k}$								Total	%
			Prob =	Anticipation step			Execution step		Relaxation step			
			0	1	2	3	1	2	1	2		
$p_{n,k}$	Prob = 0		**7897**	299	317	137	140	145	106	104	9145	86.4%
	Anticipation step	1	614	**1753**	249	100	93	74	33	34	2950	71.2%
		2	452	144	**1779**	249	192	60	40	34	2950	
		3	265	67	154	**1805**	419	113	62	65	2950	
	Execution step	1	200	41	96	103	**1984**	338	111	77	2950	79.2%
		2	145	17	34	39	338	**2010**	265	102	2950	
	Relaxation step	1	96	24	23	20	78	421	**2076**	212	2950	81.9%
		2	104	12	7	11	50	220	397	**2149**	2950	
Total			9773	2357	2659	2464	3294	3381	3090	2777	**29795**	100%
Percentage			80.8%	84.2%			70.0%		82.4%		100%	**72%**

In Table 1, the accuracy of the proposed model is 72% for the transition period and the adjacent period before and after the transition process. The bold and underlined values define the number of time-

frames, in which the estimate is exactly same with the observed scale in each transition step at a level of a grade. In the meantime, the highlighted cells illustrate the number of time-frames, in which the estimate is identical to the observed scale in each transition step at a level of a transition step. In the case, in which the LC probability is zero for the transition period, 7897 time-frames are well estimated by the CNN among 9145 time-frames. In this step, 86.4% of time-frames are identical to the observed. For the anticipation step, 71.2% of time-frames are not only rightly categorized by the CNN, but also 79.2% of time-frames are well classified for the execution step. In relaxation step, the CNN rightly estimates 81.9% of time-frames. According to the results, the CNN model can guarantee the high level of accuracy and the low level of computational burden to estimate LC probabilities in a scale of 1 millisecond.

In the case study, the performance of the integrated stochastic car-following model were examined on a variety of trajectory data of lane-changer and its surrounding vehicles, including the lead vehicles and the rear vehicles in the original and the target lane. The results of the case study show that the prediction of the integrated model with deviations is almost identical to the observed trajectories of the lane-changers and the following vehicles in the initial and the target lane. In future research, we will introduce the deep learning method to select the most appropriate parameters for the specific categorized LC manoeuvres. With this method, we can develop an adaptive multi-lane stochastic car-following model with respect to the kind of LC manoeuvres. This model can be used to establish multi-lane Cooperative Adaptive Cruise Control (CACC) as well as to model the stochastic characteristics of heterogeneous vehicular platoons in multi-lane traffic environments. Full details of the model development and simulation resulst will be presented at the conference if accepted.

References

[1] Lee, S., Ngoduy, D. and Keyvan-Ekbatani, M. “Stochastic continuous car-following model Part II: lane changing dynamics using deep learning”, *Transportation Research Part B: Methodological*, under review (2018).

[2] Lee, S., Xie, K., Ngoduy, D., and Keyvan-Ekbatani, M. “An advanced deep learning approach to real-time estimation of lane-based queue lengths at a signalized junction”, *Transportation Research Part C: Emerging Technologies*, under review (2018)

[3] LeCun, Y., Bengio, Y. “Convolutional networks for images, speech, and time series”, *The handbook of brain theory and neural networks* (1995).

Reliable Parcel Routing Policy in a Physical Internet

Ido Orenstein and Tal Raviv
Department of Industrial Engineering
Tel Aviv University
Email: ido.orenstein1986@gmail.com

1 Introduction

In the small parcel delivery industry, the "last mile cost" accounts for a significant share of the total costs. The idea of saving on this cost by handing the parcels to their recipients through automated service points (SPs) is a common practice [3]. We introduce a logistic model for the delivery of parcels to SPs that are used as drop off, pickup and intermediate storage locations. A parcel may be carried from its origin to its destination in several legs via several possible intermediate SPs. Such a system constitutes a physical internet (PI) service network [4]. The PI network is a generalization of the current practice of using a hierarchical network where the parcel can switch vehicles only in a large sorting facility (hub), and an SP is served by a single route. For the design of hierarchical service networks see [1], [2]. The PI service network topology presents an opportunity to improve metropolitan service networks by reducing the total distance that the parcels are carried, while still exploiting the possibility of shipment consolidation. In addition, such a system may save a significant amount of resources that are associated (and tied for an extended period) with the construction and operation of a large sorting facility. A related idea that is based on crowd-sourcing, i.e., with delivery of parcels by random vehicles, rather than on a planned and fixed service network, was recently introduced in [5].

In this abstract, we focus on one important operational aspect of a PI delivery network, namely, the routing of the parcels. Our model considers a service network with given locations and capacities of the SPs as well as (fixed) routes and schedules of the couriers. Our goal is to optimize the routes of the parcels within the network. As a benchmark, we consider a traditional hierarchical service network with the same set of SPs, the same amount of transportation resources and an uncapacitated centrally located hub. We are interested in the online version of the problem where parcels with different origins and destinations arrive at the system following some stochastic process. A solution to this problem is a policy by which parcels are picked up and dropped off by the couriers at the SPs. We present a policy that makes use of central information and routes each parcel through the network, so as to minimize its shipping time. The route of each parcel is obtained by a solution of the shortest path problem on a time-expanded graph that describes the current and future states of the system. A special trait of this policy is that all the resources that are needed to accomplish the delivery of each parcel are reserved upon its arrival, and thus, the system may provide reliable information on the delivery time of each parcel in advance. Such information is valuable for the shippers and the recipients. This policy is implemented for both the PI and the hierarchical systems, and a simulation is used to evaluate its performances under the two topologies.

2 Problem Definition

The problem is defined by the following input: a set of capacitated SPs, a distance matrix between the SPs, and a set of fixed tours (circular routes) that constitute the service network. Identical capacitated

couriers travel along the tours. The tour of each courier and his location along it at the beginning of the planning horizon are given. The travel time and stopping time at each SP are assumed deterministic, and thus, the arrival times of the couriers at each SP during the planning horizon can be deduced from this information. Parcels of identical dimensions arrive at the system according to a known stochastic process. Each parcel is characterized by an origin and a destination that are drawn from some known joint distribution and by its priority class. When a parcel arrives at the system, it can be admitted to its origin SP, if it has some available capacity, or rejected. Parcel rejection is at the discretion of the operator and may occur even if the SP is not at full capacity at the parcel arrival moment. After the parcel reaches its destination SP, it is collected by the recipient and the capacity it occupies in this SP is released. The time between the arrival of the parcel and its pickup is random but bounded from above. The parcel routing problem is to find a policy for pickup and drop off with the following two objectives: minimizing the expected delivery time and the expected number of rejected parcels at each class. This is a multi-objective optimization problem with two objectives for each priority class.

Two of the hardest assumptions of the model described above are as follows: (1) the deterministic travel times (2) and the deterministic and fixed service times at the SPs. To somewhat soften these assumptions, we introduce the notion of *buffer time,* which represents a specified period after the arrival of each parcel to an intermediate SP during which the parcel cannot be scheduled to be picked up by another courier.

We define a routing policy to be *reliable* if, under the deterministic travel and service time assumptions, the exact delivery time of each admitted parcel can be determined upon its admission. In this study, we focus on reliable policies only, although it is clear that the reliability requirement may come at the cost of higher rejection rate and longer delivery times.

3 Methodology

In this section, we define a myopic routing policy that is not necessarily efficient. However, it satisfies the reliability requirement. Moreover, the proposed policy is locally optimal from the perspective of each parcel and is based on the information that is available to the operator at its arrival time. This policy is applicable also to the hierarchical network, and thus, we can use it to compare the two topologies.

The current and future states of the system are represented (and maintained) by time expanded directed graphs, one for each priority class of the parcels. Each planned arrival of a courier at an SP is referred to as an event. For each event, we define a pair of nodes in the graph, referred to as a route-node and a storage-node. The arcs in the graph are as follows: *loading arcs*, which connect each storage-node to the route-node of the same event; *storage arcs*, which connect each storage-node to the storage-node of the next event in the same SP; *route arcs*, which connect the route nodes and represent the tours of the vehicles and their schedule; and finally, the *buffer-arcs,* which connect each route-node to the storage-node of the earliest event at the SP that occurs at least a buffer-time later. Each arc in the graph is associated with two properties, i.e., length and remaining capacity. The length of each arc is the time difference between its start and end nodes. The remaining capacity property represents the maximum number of additional parcels of the corresponding priority class that can be assigned to the respective resource. In the case of route-arcs, this represents the available capacity of the vehicle, and in the case of the storage-arcs, this is the available (and reservable) capacity at the SP during the epoch between the two events. The buffer-arcs are not associated directly with capacity, but their utilization is accounted

for by their parallel storage-arc(s). The capacities of the route-arcs (resp., storage-arcs) of the highest priority class network are initiated with the capacity of the corresponding vehicle (resp., corresponding SPs). The remaining arc capacities of the lower priority graphs are initiated with smaller values, where the difference represents the extra capacity that is reserved for the use of the higher priority classes only. The initial capacity for each priority class should be determined by the planner according to the importance attributed to the priority classes. However, in this work, we assume that these capacities are given.

Whenever a parcel arrives at the system, a shortest path from the previous storage node in its origin to the earliest possible storage-node in its destination on the time-expanded graph of its priority class is calculated. Only arcs with positive remaining capacity are considered and only destination nodes with positive remaining storage capacity for the period allowed for pickup are considered. If no such path exists, or if its length is deemed unacceptable by the planner, the parcel is rejected. Otherwise, the path represents the set of resources (capacity of the vehicles and SPs along the route and until the latest allowed pickup time) that are needed to transfer the parcel. These resources are reserved for the parcel, and the recipient is notified about the planned arrival time. The remaining capacity of the arcs along the path and during the allowed pickup period in all the priority class graphs are decremented by one. Note that this may lead to negative capacity values at some arcs in the lower priority class graphs but not in the highest one. When a parcel is picked up by the recipient before its latest allowed pickup time, the storage capacity for the remaining time that was reserved for it is released and the remaining capacity on the storage arcs is incremented accordingly.

While the myopic parcel routing policy described above is optimal when the capacity constraints of the vehicle and the SP are unbinding, the policy is too short sighted if this is not the case. Indeed, parcels that arrive earlier may congest resources that may be more beneficial later. For example, if the shortest path of a parcel on the time expended graph utilized the last capacity unit of an arc (vehicle or storage) but the second best path is only slightly longer and require only arcs with a lot of spare capacity, it may be reasonable to route the parcel via the second best route. By doing so, we leave the nearly congested resource available to other parcels that will arrive later and may save much more from using it in terms of delivery time. We propose a heuristic method that will divert parcels from congested resources, if this can be done without causing long delays, by imposing a "congestion fee" on arcs that are utilized nearly at their capacity. The fee is determined by the following piecewise linear function of the resource remained capacity ratio u (i.e., current remained resource capacity / original capacity).

$$f(u) = \begin{cases} 0, & u \geq 1-\alpha \\ \left(1 - \dfrac{u}{1-\alpha}\right)(\beta - 1)l, & u < 1-\alpha \end{cases}$$

With $0 < \alpha < 1$ but typically close to 1 and $\beta \geq 1$. The shortest path is calculated on a graph where the length of each arc along the path is considered as the sum of the initial length l of the arc (the time difference between nodes) and its $f(u)$ value. The routing policy with fee is clearly an extension of the myopic one. The latter is a special case with $\beta = 1$.

4 Numerical experiment

In this section, we present a sample of the results obtained in our numerical study. We created a simulation environment where parcels arrive at the SPs according to a Poisson arrival process and are

routed using our algorithm by two sets of scheduled tours of vehicles. One represents a hierarchical service network in a favorable setting, and one represents a simple PI service network. The two networks consist of a 20×20 grid with SPs that are located at equal distances of a five-minute drive from each other. This geography is equivalent in size to a relatively large metropolitan area with a dense coverage of service points. For the hierarchical service network, the location of the depot coincides with the location of one of the SPs in the center of the grid. The hierarchical network is served by 40 tours that start and end at the hub and visit 10 SPs each. The tours are served in a round robin fashion by 40 couriers, and each SP is visited by a courier exactly every 3.5 hours. The PI service network consists of 40 tours each served by a single vehicle. Twenty tours run back and forth along the south-north lines of the grid and twenty along the east-west lines. The location of each vehicle at the beginning of the planning horizon was selected randomly. The total cycle time of each tour in the PI is 6.5 hours. The service time at each SP was assumed five minutes for the PI network and four minutes for the hierarchical network. We set the service time to be shorter for the hierarchical network since the amount and complexity of the work in this setting is slightly lower. The service time of the vehicle in the hub is 25 minutes, since the task of fully unloading, loading and sorting the parcels is more time-consuming. The buffer time was set to 5 minutes in both systems, i.e., a parcel can be sent on a different vehicle five minutes after the vehicle that dropped it off left the SP or the hub. The parcels arrived at a rate of 50 parcels/day to each SP, and their destinations were selected randomly. In total, this represents a rate of 20,000 parcels/day. The parcel pickup time by the recipient was drawn from U(0,12) hours. We considered only a single priority class. The capacity of the SPs was set to 100 parcels which was found to be sufficient, and the hub was uncapacitated.

We tested two levels for the capacity of the vehicles, namely, 100 and 130 parcels. Under these conditions, both systems exhibited stable behavior and reached a steady state after a few days of simulation, as opposed to cases with lower capacity. The simulation was run for 40 days (excluding warmup times), and no parcel rejection was observed. In addition, we ran the simulation without capacity constraints on the vehicles and SPs to explore the potential of both topologies when resources are abundant.

Both service network topologies were tested under the myopic policy and the myopic policy with congestion fee with various values of α and β. In the table we present the case of $\alpha = 0.9$ and $\beta = 4$ but very similar results were obtained for other values of $\alpha \in [0.7,0.95]$ and $\beta = [2,5]$ which demonstrate the robustness of the congestion fee idea.

Vehicle Capacity	SP Capacity	Hierarchical Myopic	Hierarchical congestion fee	PI Myopic policy	PI congestion fee
100	100	7:29	7:27	7:27	6:54
130	100	7:25	7:25	6:36	6:31
Unbinding	unbinding	7:25	7:25	6:28	6:28

In the table, we present the average parcel delivery time in hours and minutes for both topologies under some capacity conditions and routing policies. It is apparent from the table that when using a sufficient amount of transportation and storage resources, the PI service network allows shorter delivery times than the traditional hierarchical one even under the myopic policy. However, when the

transportation resources are scarce, the advantage of the PI topology diminishes. The extended myopic policy with congestion fees can be used to mitigate the effect of resources scarcity in the PI network. The Hierarchical network gains nothing in terms of delivery time from the extended policy. This can be explained by the fact that in such a network each parcel has only one possible path on the physical network. Thus, any path diversion on the time-expended graph requires causing significant delays in the delivery.

5 Conclusions

We presented a routing policy for the delivery of parcels in a metropolitan area that can provide reliable information on the delivery time in advance. We demonstrated that this policy performs better in a PI service network topology than in the traditional hierarchical one. We note that the PI topology does not require the expensive construction and operation of urban sorting facilities and may offer a robust and economical method to deliver parcels. The method should be tested and tuned with different service networks, and resource capacities. A method to design an effective service network that operates under such a parcel routing policy is an interesting topic for future research.

References

[1] Alumur, S. and Kara, B.Y., 2008. Network hub location problems: The state of the art. *European journal of operational research, 190(1)*, pp.1-21.

[2] Çetiner, S., Sepil, C. and Süral, H., 2010. Hubbing and routing in postal delivery systems. *Annals of Operations Research, 181(1)*, pp.109-124.

[3] Faugere, L. and Montreuil, B., 2017. Hyperconnected Pickup & Delivery Locker Networks. Proceedings of 4th International Physical Internet Conference, Graz, Austria

[4] Montreuil, B., 2011. Toward a Physical Internet: meeting the global logistics sustainability grand challenge. *Logistics Research, 3(2-3),* pp.71-87.

[5] Tenzer, E.Z., and Raviv T., 2018. Crowd-shipping of small parcels in a physical internet, working paper, Tel-Aviv University

Finding critical links to estimate a Macroscopic Fundamental Diagram in congested urban networks

Elham Saffari

School of Civil Engineering

University of Queensland

Mehmet Yildirimoglu (corresponding author)

School of Civil Engineering

University of Queensland

Email: m.yildirimoglu@uq.edu.au

Mark Hickman

School of Civil Engineering

University of Queensland

Introduction

Traffic congestion has been increasing due to population growth and rapid development of urban areas. Thus, proper traffic models and monitoring plans are essential, especially for populated cities. The Macroscopic Fundamental Diagram (MFD) has been recently described by Geroliminis and Daganzo (2008) and offers promising results for traffic monitoring and control purposes. Essentially, the MFD studies the relationship between average flow and average density across relatively homogeneous urban areas.

The primary idea of the Macroscopic Fundamental Diagram was presented by Godfrey (1969), and later investigated by Herman and Prigogine (1979) and Mahmassani et al. (1984). Using real data collected from Yokohama, Geroliminis and Daganzo (2008) experimentally showed that the MFD exists at an urban scale. This study showed that a homogeneous urban region (with limited variance of link density) can be modelled with the MFD, which provides a unimodal, low-scatter and demand insensitive relationship between average density and average flow.

In addition to studies considering the effective factors on MFD shape, many studies have been trying to estimate the MFD either with real data or simulation data. In a recent study, Ambuhl and Menendez (2016) proposed a fusion algorithm, using both the loop detector and floating car data, to estimate the MFD. Ortigosa et al. (2014) proposed a quasi-optimal search algorithm in order to find the best set of links in order to estimate the most accurate MFD, but using only a limited number of links. Zockaie et al. (2018) recently developed a mathematical model to find the optimal location of measurement points to estimate the MFD in a large and heterogeneous network. A limitation of this model is that it incorporates the ground-truth MFD in the modelling framework, which is rarely available in real networks.

Given that the monitoring resources (e.g. loop detectors, probe vehicle data, etc.) are limited in real-world networks, acquiring adequate data to estimate the MFD is very important. Therefore, this

study aims to identify the critical links where loop detectors should be installed to estimate the MFD and minimize the estimation error between the MFD that is estimated by limited measurement points (i.e., critical links) and the "true" MFD. [In this abstract, we refer to the MFD which is estimated from all the links as the true MFD.]

Methodology and Results

In this research, we assume a network with no loop detectors in any of its links, and we aim to identify the critical links where mid-block loop detectors should be installed to estimate an accurate MFD. We also assume probe vehicle data (position and speed) with a given penetration rate and measurement frequency is available (further details are given below). While we use a true MFD (estimated using loop detector data from all the links) for evaluation purposes, we target a final solution method that only relies on the probe vehicle data as the ground truth.

The network that we use for this study is the network of Barcelona, which is modelled in Aimsun, a microscopic simulation package. Employing the Aimsun API, therefore, we collect probe vehicle data within a 1.5-hour simulation. In order to get a better representative average speed, we collect probe vehicle data every 5 seconds. Randomly selecting 10 percent of vehicles, we aggregate the data in every minute and calculate link average speeds. Figure 1 shows the distribution of link average speeds at several time periods throughout the simulation. We clearly see that, as time progresses, the average speed distribution shifts to the left, the number of links with low speed increases, and the network gets more congested.

Let $Q(t)$ and $K(t)$ denote the average flow and average density weighted by the link lengths where the detectors are located, respectively. To derive MFDs (both the true MFD and the estimated MFDs), we apply Eqs. (1) and (2) and calculate $Q(t)$ and $K(t)$.

$$Q(t) = \frac{\sum_{i=1}^{I} q_i(t) \cdot l_i}{\sum_{i=1}^{I} l_i} \qquad (1) \qquad\qquad K(t) = \frac{\sum_{i=1}^{I} k_i(t) \cdot l_i}{\sum_{i=1}^{I} l_i} \qquad (2)$$

where I is the set of links where detectors are installed, l_i is the length of link i, and $q_i(t)$ and $k_i(t)$ are the flow and density on link i at time step t, respectively. Note that the true MFD uses all the links in the network, while the estimated MFD results from a subset of the links.

We assume "critical" links are the links that represent average traffic conditions in the network. Instead of collecting data from the entire network in order to capture the average traffic conditions, one can choose to collect data from a limited number of links yet capture a fair amount of variability. To this end, we apply Principle Component Analysis (PCA) in order to reduce the dimension of the data as well as to detect the critical links. PCA is considered to be one of the most common unsupervised learning algorithms and the most popular dimensionality reduction algorithm. The goal of PCA is to transform the original variables into a few interpretable linear combinations of them, which are in turn called principle components (PCs). Thus, it reduces the dimensions of a d-dimensional dataset by projecting it onto a k-dimensional subspace while maintaining the most variance in the original data (where $k < d$). In order to interpret the contribution of the original variables into each PC, we need to calculate the loading matrix. Basically, loadings estimate the correlation between an original variable and a PC, which is a proxy for the information they share. We, therefore, use the links with the highest loading values to identify the critical links in our framework.

As a result of our analysis, we see that 30 PCs explain 85 percent of the variance in the original dataset (see Figure 2). The first and the second PCs explain 29 percent and 8 percent of the variance, respectively.

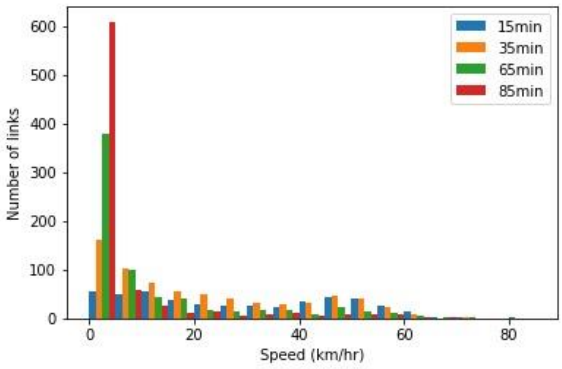


Figure 1: Distribution of average speed throughout the network in four different simulation periods

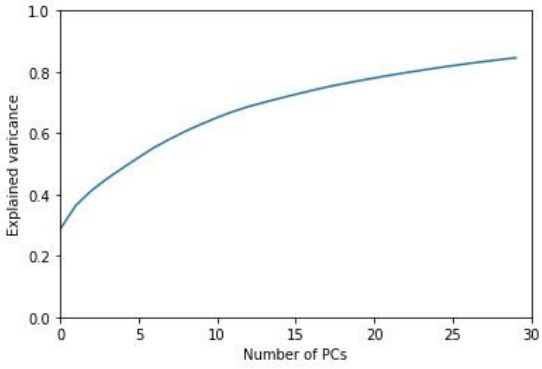


Figure 2: Cumulative explained variance by PCs

As a significant part of the variance is explained by the first PC, we first attempt to identify critical links using only the features resulting from this first PC. We find the first 10, 20, 30 and 40 links that are most correlated with the first PC and calculate the weighted average flow and density using only the loop detector data from them. As shown in Figure 3, selecting more links leads to a more accurate and less scattered MFD. Yet, none of the configurations produces flow values as high as the true MFD.

To explore the effects of multiple PCs, we choose the 20 highest-contributing links from PC1 and PC2. Figure 4 depicts the MFDs derived from these 20 most contributing links to PC1 and PC2, separately. Clearly, the contributing links to PC2 are less congested and carry less flow; therefore, they do not provide a good approximation to the true MFD. Taking into account both PC1 and PC2, as represented in Figure 5, we can observe that the estimated MFD from PC1 is a better estimation than the MFD from the mixed PCs (consisting of 30 links from PC1 and 20 links from PC2). Figure 6 shows the location of selected links from PC1 in the network. We can see that the critical links are not selected only from one particular part of the network; that is, they capture the traffic state from several different parts of the network.

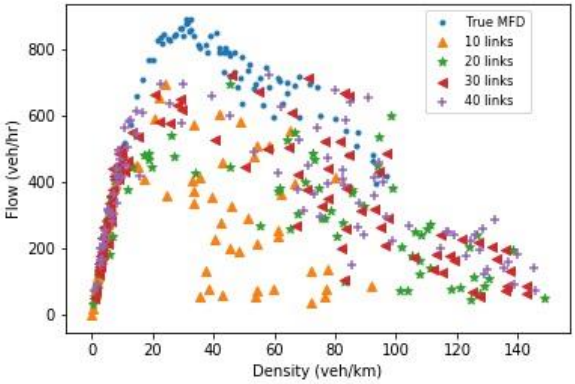


Figure 3: Comparison of the true MFD and MFDs estimated with the selecting links from the first PC

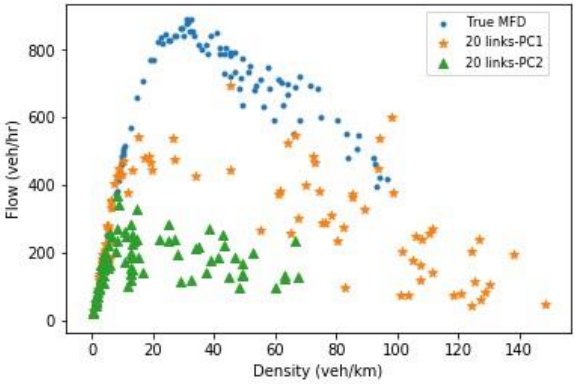


Figure 4: Comparison of the estimated MFDs using PC1 and PC2

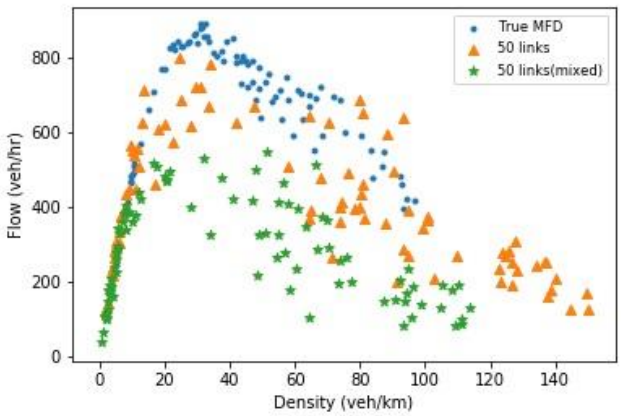


Figure 5: Comparison of the estimated MFDs using PC1 and mixed PCs (PC1 and PC2)

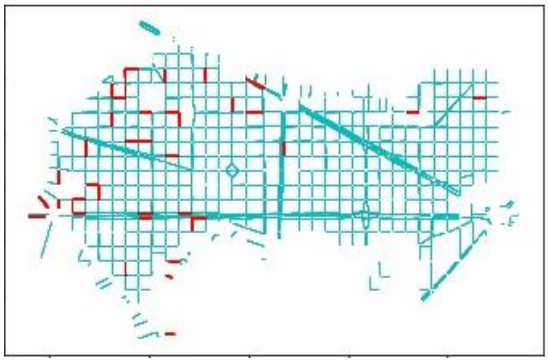


Figure 6: Selected links from PC1

While it is still not clear how one can design a proper combination of PCs and associated links, the results demonstrate that the proposed method has the potential to develop an unsupervised framework to identify the critical links for the estimation of the MFD. Selecting the highest-contributing links to PC1 shows a more accurate MFD since the first PC explains a greater variance (i.e., more congested links and more flow) in comparison to other PCs. Here, we only use PC1 and PC2 for our estimations; however, due to PCs being statistically uncorrelated, each of them represents different states of the network. For the future investigations, we will consider a higher number of PCs, thereby taking into account more variability in the network.

References

Ambühl, L., & Menendez, M. (2016). Data fusion algorithm for macroscopic fundamental diagram estimation. Transportation Research Part C: Emerging Technologies, 71, 184-197.

B Godfrey, J. W. "The mechanism of a road network." Traffic Engineering & Control 8.8 (1969).

Geroliminis, Nikolas, and Carlos F. Daganzo. "Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings." Transportation Research Part B: Methodological 42.9 (2008): 759-770.

Herman, Robert, and Ilya Prigogine. "A two-fluid approach to town traffic." Science 204.4389 (1979): 148-151.

Mahmassani, Hani S., James C. Williams, and Robert Herman. "Investigation of network-level traffic flow relationships: some simulation results." Transportation Research Record 971 (1984): 121-130.

Ortigosa, Javier, Monica Menendez, and Hector Tapia. "Study on the number and location of measurement points for an MFD perimeter control scheme: a case study of Zurich." EURO Journal on Transportation and Logistics 3.3-4 (2014): 245-266.

Zockaie, Ali, Meead Saberi, and Ramin Saedi. "A resource allocation problem to estimate network fundamental diagram in heterogeneous networks: Optimal locating of fixed measurement points and sampling of probe trajectories." Transportation Research Part C: Emerging Technologies 86 (2018): 245-262.

A Study on Driver's Stopping Behavior Focusing on Generalization

Hirotoshi SHIRAYANAGI
Graduate School of Science and Engineering
Ehime University, 3 Bunkyo-cho, Matsuyama, Japan
Email: shirayanagi@cee.ehime-u.ac.jp

Takahiro TSUBOTA
Graduate School of Science and Engineering
Ehime University

Shinya KURAUCHI
Graduate School of Science and Engineering
Ehime University

Toshio YOSHII
Graduate School of Science and Engineering
Ehime University

1 Introduction

1.1 The difference in driving behavior between real space and virtual space

Driving simulators are used for research purposes in the area of human factors to monitor driver behavior, performance and attention and in the car industry to design and evaluate new vehicles or new advanced driver assistance systems. However previous studies (Blaauw, 1982; Godley et al., 2002) have suggested that there is a difference in driving behavior between virtual space and real space, and this difference includes individual differences. It is necessary to clarify the cause of individual difference on driving behavior measured by a driving simulator.

1.2 Acquired behavior and generalization

Most of behavior in daily life is acquired behavior that is something persons discover through trial, error and observation. Previous study (Flora, 2004) has suggested that acquired behavior is triggered by positive reinforcement that is stimulus encourages a certain acquired behavior. Once persons have been trained to respond to a certain positive reinforcement, the positive reinforcement may produce the same acquired behavior in any situation. This phenomenon is called generalization (Paivio, 1971).

1.3 Acquired driving behavior and generalization

It has been suggested that hierarchical model explained driving behavior (Keiskinen, 1996). This model described the driving behavior into four stages that are a basic skill stage, a stage of operation in a certain traffic condition, a motivational aspect stage and an attitudinal aspect stage, and drivers get driving behavior of upper stage by accumulating driving experience. Driving behavior is a type of acquired behaviors, and therefore focusing on lower two hierarchies, it is possible that drivers with shorter driving experience drive based on the cue of first stage that is considering only the own vehicle as positive reinforcement, and drivers with longer experience drive based on the cue of second stage that is considering following vehicles and fellow passengers as positive reinforcement. Furthermore it is possible that a procedure in driving is generalized by accumulating driving experience. This study aims to relate the generalization of driver's stopping behavior to the driver's driving experience by comparing the stopping behavior in real space and virtual space presented by a driving simulator.

2 Method

2.1 Definition of positive reinforcement for driver's stopping behavior

Two types of positive reinforcement for driver's stopping behavior are defined. One is duration time required to stop. It is formulated as presented in equation (1). It is assume that this positive reinforcement is processed for drivers with shorter driving experience, because this stopping behavior is an action that takes into consideration driving only own vehicle.

$$t = d/v \tag{1}$$

where

t : duration time from driver puts on the brakes to vehicle stops [s],

d: distance from the point driver puts on the brakes to the point vehicle stops [m], and

v: velocity that driver puts on the brakes [m/s].

Another is deceleration required to stop. It is formulated as presented in equation (2). It is assume that this deceleration is processed for drivers with longer driving experience, because this stop behavior is an action that takes into consideration comfortable for following vehicles and passengers.

$$a = v^2/d \tag{2}$$

where

a: deceleration required to stop $[m/s^2]$,

v: velocity that driver puts on the brakes [m/s], and

d: distance from the point driver puts on the brakes to the point vehicle stops [m].

Therefore the relationship between velocity that driver puts on the brakes and distance from the point driver puts on the brakes to the point vehicle stops is formulated as presented in equation (3). It is possible to consider positive reinforcement for each driver by estimating the value of the parameter β, and it is also possible to consider the generalization of driving behavior by comparing the difference in the value of the parameter β between real space and virtual space. If drivers with longer driving experiment

perform stopping behavior based on deceleration required to stop as positive reinforcement, it is expected that the value of parameter β of driving data in the real space close to 2, and if their stopping behavior is generalized, it is expected that the value of parameter β is the same value regardless of the situation: real space or virtual space. On the other hand, drivers with shorter driving experiment perform stopping behavior based on duration time to stop as positive reinforcement, it is expected that the value of the parameter β in the real space close to 1, and if their stopping behavior is not generalized, it is expected that the value of parameter β is different from real space and virtual space.

$$d = \alpha v^{\beta} \tag{3}$$

where

d: distance from the point driver puts on the brakes to the point vehicle stops [m],

v: velocity that driver puts on the brakes [m/s], and

α, β: unknown parameter.

2.2 Participants and driving conditions

Four drivers gave their informed consent to participate in the experiment. One driver has been passed 12 years (Driver-12), two drivers have been passed three years (Driver-3a, 3b) and one driver has been passed one year (Driver-1) since they got their driver's licenses. Participants were instructed to drive and stop following the instruction of experimenter in both real space and virtual space presented by a driving simulator.

3 Results and Discussions

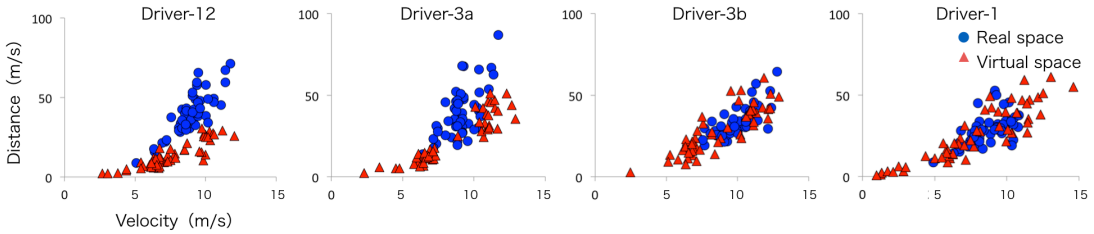


Figure1: The relationship between velocity that driver puts on the brakes and distance from the point driver puts on the brakes to the point vehicle stops.

Table1: The values of parameter in the real space and virtual space.

	The data in real space (n=51)				The data in vitual space (n=54)			
	α	(t-ratio)	β	(t-ratio)	α	(t-ratio)	β	(t-ratio)
Driver-12	0.52	(2.77)	1.88	(12.1)	0.18	(1.92)	2.08	(9.00)
Driver-3a	0.32	(1.21)	2.07	(5.84)	0.24	(3.17)	2.02	(14.9)
Driver-3b	0.62	(1.38)	1.65	(5.45)	0.93	(3.36)	1.50	(12.3)
Driver-1	1.84	(2.36)	1.20	(6.44)	1.30	(4.05)	1.37	(13.6)

β is not significantly different from 2

β is not significantly different from 1

Figure 1 shows the relationship between velocity that driver puts on the brakes and distance from the point driver puts on the brakes to the point vehicle stops for each participant. The values of parameter

are summarized in Table1. The result shows that the value of the parameter β of driver-12 and driver-3a estimated from the data driving in real space is not significantly different from 2. This suggests that drivers perform stopping behavior based on deceleration required to stop as positive reinforcement. The value of the parameter β estimated from the data driving in virtual space is not significantly different from 2. This suggests that they performed stopping behavior based on the same positive reinforcement regardless of the situations: real space or virtual space, which suggests that their stopping behaviors are generalized. On the other hand, the result of driver-1 shows that the value of the parameter β estimated from the data driving in real space not significantly different from 1. This suggests that driver performs stopping behavior based on duration time required to stop as positive reinforcement. The value of the parameter β estimated from the data driving in virtual space is significantly different from 1 and 2. This suggests that stopping behaviors is not generalized, because the driver doesn't have enough driving experience. The result of driver-3b shows that the value of the parameter β estimated from the data driving in real space is not significantly different from 2 and the value of the parameter β estimated from driving in virtual space is significantly different from 1 and 2. It is possible that 3 years of driving experience is a transition period of generalized stopping behavior, although it is necessary to consider the validity of the interpretation, because there are a few participants in this study.

4 Conclusion

This study investigates the relationship between driving experience and generalization of driver's stopping behavior. The results showed that drivers with longer driving experience performed stopping behavior based on the same positive reinforcement regardless of the situations: real space or virtual space, which suggests that their stopping behavior is generalized. On the other hand, drivers with shorter driving experience did not exhibit the generalization in their behavior.

Acknowledgement

This work was supported by committee on advanced road technology of the ministry of land infrastructure, transportation and tourism.

References

[1] G.J. Blaauw, "Driving experience and task demands in simulator and instrumented car: a validation study", Human Factor, 24 (4), pp. 473-486, 1982.

[2] Stuart T. Godley, Thomas J. Triggs and Brian N. Fildes, "Driving simulator validation for speed research", Accident Analysis & Prevention, 34 (5), 589-600, 2002.

[3] Stephen R. Flora, "The Power of Reinforcement", Albany: State University of New York Press, 2004.

[4] Paivio, A., "Image and verbal processes", New York: Holt, Rinehart and Winston, 1971.

[5] Keskinen, E., "Why do young drivers have more accidents? Junge Fahrer Und Fahrerinnen. Referate der Esten Interdiziplinären Fachkonferenz 12–14 Dezember 1994 in Köln. Berichte der Bundesanstalt fur Strassenwesen. Mensch und Sicherheit, Heft M 52, 1996.

Operations Design for High-velocity Intra-city Package Service

Iman Dayarian

Culverhouse College of Business, Information Systems, Statistics and Management Science

University of Alabama

Adolfo Rocco Rocco **Alex M. Stroh** **Martin W.P. Savelsbergh**

Alejandro Toriello

Alan L. Erera

School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, Georgia, USA

Email: alan.erera@isye.gatech.edu

1 Problem Setting

We consider an operations network design problem for package courier systems operating high-velocity services within large urban areas. High-velocity services include standard next-day services where packages collected today are delivered tomorrow and also same-day service where pickup and delivery occur on the same day. Demand for high-velocity services is growing. In this work, we collaborate closely with one of the largest package couriers in China; their business model includes a plan to grow high-velocity services within Chinese megacities with a new operating model. We build novel optimization technology to configure vehicle operations using new and novel *rate-based* routing and network design models that use parcel demand rates per time as inputs, and that determine both route capacity and service-level feasibility with vehicle flow rates per time between locations induced by repeated execution of vehicle routes during an operating day.

Consider a system with a number of small hub terminals throughout an urban area. These locations, denoted *local hubs*, are used for consolidation of packages into and out of a set of small geographic service regions into which the urban area has been partitioned. Due to the congested urban environment, the *couriers* who pick up and delivery goods directly from and to customers do not operate large vehicles; instead, they walk or use small delivery bikes with limited package capacity. Many couriers operate within each local hub service region, but they do not visit the local

hub and instead spend the day working within their assigned *unit zone*. Packages are transferred to and from couriers within their unit zones via a fleet of small-capacity transfer vehicles known as *riders*. Riders transfer packages with a courier at a designated *access hub* location either synchronously via timed meet-ups, or asynchronously through the use of parcel lockers.

Packages are transported between service regions of different local hubs via second set of vehicles, known as *shuttles*. Shuttle vehicles are larger than rider vehicles, since they only need to stop at local hub locations. In a large urban area characterized by many service regions and local hubs, it is likely not economical to schedule direct shuttle movements between all pairs of local hubs. Packages can be cross-docked between shuttles at local hubs to enable non-direct service. Overnight storage of parcels is not allowed at local hubs, and instead only at a small set of larger facilities denoted *gateway hubs* that also provide intercity service for packages moving into or out of the urban area. Shuttle services thus also transfer outbound intercity packages from local hubs to gateway hubs, and inbound packages from gateway hubs to local hubs.

In this research, we consider approaches to design rider and shuttle vehicle operations, and associated cross-dock transfers, to enable effective intra-city transfer of packages. The objective is to create a design that moves packages between origins and destinations meeting timing requirements, while minimizing the cost of providing the services. Unlike traditional approaches for city logistics design (see excellent examples in [1] and [2], and the review in [3]), we seek to construct repeatable service cycles for both shuttles and riders that can be executed during (a portion of) the operating day to provide continuous transfer service.

2 Service Network Design Problems

Consider a multigraph $\mathcal{G} = (\mathcal{U}^G \cup \mathcal{U}^L \cup \mathcal{U}^A, \mathcal{A})$ with the node set representing the union of gateway hubs (GH), local hubs (LH), and access hubs (AH) respectively and arc set $\mathcal{A}$ representing (directed) transportation connections between nodes. Every package origin or destination in the service region is uniquely served by a courier who meets a rider at a single AH, thus we map demand to access hubs. In this work, each AH is served by a unique LH, and each LH is served by a unique GH. A heterogeneous fleet of vehicles provides transport service, where Q^v is the capacity of vehicle type v. The company provides a number of different service classes to customers, including same-day (SD), next-morning (NM), and next-evening (ND), and each leads to corresponding deadlines. Only SD packages are transferred directly from an origin LH to a destination LH on the same day. The next-day intracity classes travel to a gateway hub on the pickup day and then from a (possibly different) gateway hub on the delivery day. We do not consider the simple operations required to transfer packages overnight between the small set of gateway hubs. Intercity classes always originate from or are destined to a gateway hub.

2.1 Rider scheduling

We will illustrate the primary ideas of rate-based models using *rider scheduling* pickup-and-delivery routing models. First assume that rider and shuttle operations are operated by separate vehicles, and that each local hub operates an independent rider fleet serving its access hubs. Each access hub is served by a single rider route, perhaps with multiple vehicles assigned. Our goal is to design repeatable routes for riders, each beginning at the LH and visiting some subset of its access hubs, that meet service requirements with low cost. In this talk, assume that inbound packages arrive at the LH for delivery to AH i at a constant flow rate of q_i^I parcels per time. Similarly, outbound packages collected by the courier at AH i are generated at flow rate q_i^O.

Travel along arc $a \in A$ requires ℓ_a time, and a stop time of t_A is required at each AH to deliver and pickup packages and a stop time of t_D is required at the LH. Then, a rider visiting access hubs $R = \{1, 2, ..., |R|\}$ in sequence from LH would require time duration of $\ell_R = \sum_{a \in a(R)} \ell_a + |R|t_A + t_D$ for each circuit, where $a(R)$ includes the arcs $(LH, 1)$ and $(|R|, LH)$ plus the consecutive AH connections. The headway between visits to an AH is given by $H_R = \frac{\ell_R}{m_R}$ if m_R riders execute route R with equal headways. The average *waiting time* for a package served by route R is $\frac{H_R}{2}$.

Suppose that the allowable rider transfer time for an inbound or outbound package is T. Then route R with m_R assigned riders is *service feasible on average* if the inbound packages on average arrive on time at the (most constrained) AH $|R|$,

$$\frac{H_R}{2} + \sum_{a \in a(R) \setminus (|R|, LH)} \ell_a + (|R| - 1)t_A \leq T \quad ,$$

and the outbound packages arrive on time at the LH from the (most constrained) AH 1,

$$\sum_{a \in a(R) \setminus (LH, 1)} \ell_a + (|R| - 1)t_A + \frac{H_R}{2} \leq T \quad .$$

Note a more conservative model can increase the waiting time from the average; note that the maximum waiting time is simply H_R.

Routes must also provide enough parcel transfer capacity. Given a rider vehicle size of Q_R, route R with m_R riders assigned provides $m_R Q_R$ parcels per time on each route leg $(LH, 1), (1, 2), ..., (|R|, LH)$. The following route leg constraints then ensure that the route provides enough physical capacity:

$$\sum_{j=i+1}^{|R|} q_j^I + \sum_{j=1}^{i} q_j^O \leq m_R Q_R \quad \forall\, i \in \{0, 1, ..., |R|\}$$

Let m be the smallest number of riders that can feasible execute route R, and let m be the cost of selecting route R. We build a set partitioning (covering) model with binary decision variables x_R to minimize the total number of riders, subject to constraints that ensure that each AH is included on exactly (at least) one route. Note that as a route R contains larger numbers of access hubs, it becomes less likely to be service-time feasible. For example, consider a subset S

of access hubs and let $t(S)$ and $h(S)$ be the duration of the minimum time traveling salesperson route on $S \cup \{LH\}$ and the duration of the minimum time Hamiltonian path on $S \cup \{LH\}$ rooted at LH respectively, or good lower bounds for these durations. Then, no rider route $R \supseteq S$ is feasible if $h(S) + (|S| - 1)t_A > T$, and no execution of R with m or fewer drivers is feasible if $\frac{t(S)}{2m} + h(S) + (|S| - 1)t_A > T$. Leveraging ideas like these, it is possible to build practical solution approaches for this set partitioning model via smart complete enumeration for all AH subsets up to a maximum cardinality. We have solved models for the dozens of local hubs in a large test urban area in China, and will report computational results in the talk, including the sensitivity of costs to conservatism in estimated waiting time. We also solve a different variant of the problem that seeks to maximize a flow-weighted measure of service quality (total parcel transfer time) given a fixed rider fleet size, and will present those results as well.

2.2 Shuttle network design

The design of shuttle vehicle operations is significantly more complex than that for rider operations due to the many-to-many nature of demand and the premise that cross-dock transfers of parcels between vehicles are allowed at LH locations. We develop novel service network design optimization approaches where both the capacity and time feasibility of the design are determined by vehicle flow rates along repeatable cycles of hubs (LH and GH). Related recent work includes [4] which uses a detailed time-space network to model both consolidation timing and commodity service constraints; this paper also presents a good literature review of earlier models. Our goal in this work is to investigate a simpler mechanism for enforcing service time feasibility using flat networks that may scale better with large numbers of commodities.

In this talk, we describe the two most important phases in a sequential shuttle design problem: path selection and cycle selection. The path selection phase determines an origin-to-destination cross-dock transfer path for each *commodity*. The cycle selection phase creates repeatable shuttle cycles and assigns vehicles to these cycles to meet capacity and service requirements. Let commodity k be the parcels sharing the same origin, destination, and service requirement: $(o_k, d_k, \delta_k, q_k)$, where δ_k is the available time to transfer parcels from hub o_k to d_k, and q_k is the demand rate in packages per time.

For the path selection phase, a mixed-integer program with binary decisions x_k^p that select a single path p for each commodity k and continuous dispatch frequencies z_a^v for vehicles of size Q^v on directed arc a is developed. Path p is feasible for k if the total path travel and transfer time, plus any waiting time for dispatches, does not exceed δ_k. Waiting time along a path decreases with increasing vehicle dispatch frequencies for arcs $a \in p$. By assuming that maximum total waiting time for a path is distributed equally at each hub visited by a path, we create a set of linear constraints that ensure that dispatch frequencies on arcs a allow service requirements to be

met. Similar to the rider model, we also ensure that dispatch frequencies of different truck types provide sufficient parcel transfer capacity for all paths using arc a.

In the cycle selection phase, we again use an integer program where y_c^v is the number of vehicles of type v assigned to a cycle c that visits some local hubs and at most one gateway hub. Given the total duration (travel plus package transfer time) of cycle c operated by vehicle v, each arc $a \in c$ the inverse in dispatch frequency for each vehicle. Therefore, a linear constraint can be used to determine if a selected set of cycles and vehicle assignments given by y_c^v meets the dispatch requirement of each arc z_a^v from the path selection model; this is the primary feasibility consideration, and the objective is to minimize the vehicle costs.

Similar to the rider problem, we have solved shuttle network design problems using this methodology for a large test urban area with dozens of local hubs and a handful of gateway hubs. Again, the talk will present results that demonstrate the sensitivity of costs to conservatism in estimated waiting time, but also to the average available slack between δ_k and the minimum travel time between o_k and d_k. The utility of the proposed models will be demonstrated via this study.

References

[1] T.G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems", *Transportation Science* 43, 432-454 (2009).

[2] V.C. Hemmelmayr, J-F. Cordeau, and T.G. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics", *Computers and Operations Research* 39, 3215-3228 (2012).

[3] M.W.P. Savelsbergh and T. Van Woensel, "50th anniversary invited article?city logistics: Challenges and opportunities", *Transportation Science* 50, 579-590 (2016).

[4] T.G. Crainic, M. Hewitt, M. Toulouse, and D.M. Vu, "Service network design with resource constraints", *Transportation Science* 50, 1380-1393 (2014).

Design of urban transportation infrastructure for optimal passenger throughput

Allister Loder
Institute for Transport Planning and Systems (IVT)
ETH Zürich, Switzerland Email: allister.loder@ivt.baug.ethz.ch

Michiel C.J. Bliemer
Institute of Transport and Logistics Studies (ITLS)
The University of Sydney

Kay W. Axhausen
Institute for Transport Planning and Systems (IVT)
ETH Zürich, Switzerland

1 Introduction

We build our transportation infrastructure for carrying people and goods. However, too many vehicles on the road at the same time leads to congestion, making journey speeds unsatisfactory and increase negative external costs. At the same time, replacing cars with buses can benefit the overall flow of passengers. Therefore, how much traffic and which combination of buses and cars is optimal for a city and how should it be priced? This question is key to transport planning and has been raised since the second half of the 20^{th} century [1]. The question has been addressed in many facets, but, so far, simple and (closed) macroscopic models coupling human preferences (demand) and infrastructure investment choices (supply) in multi-modal networks have rarely been attempted, other than running many large scale simulation scenarios. However, the recently introduced macroscopic fundamental diagram (MFD) and in particular its bi-modal extension to the 3D-MFD [2, 3, 4] allows us to present a novel approach to address this gap and to improve our understanding of optimal traffic for cities.

This novel approach builds upon the network design problem [5], bus network design problem [6] and travelers choices for transportation modes [7], and we therefore speak of the 3D-MFD-Network Design Problem (3D-MFD-NDP). The idea is a bi-level optimization problem where at the upper level infrastructure choices are made to minimize the total travel time in the network and at the

lower level traffic distributes across routes following Wardrop's equilibrium principle. We account for human preferences in the equilibrium with commuters valuation of time and perception of route costs. We focus on the long-term aspects and thus do not account for network dynamics. In the following, we discuss the 3D-MFD-NDP model.

2 The Model

We consider a city consisting of K sub-regions or single MFD reservoirs. We partition networks into several sub-regions based on road network topology to obtain well-defined MFDs [8]. Each region has an infrastructure length L_k, a road network length $L_{k,car}$, an average intersection spacing l_k, a bus network length $L_{k,bus}$ of which Φ_k is dedicated, an average bus stop spacing b_k, a bus network design α_k [9], and a headway τ_k. In this study, the core design variables are limited to these variables listed, because they factor into the parametrization of the recently introduced functional form for the 3D-MFD [10]. From the 3D-MFD we then derive the travel times and speeds for the lower level user equilibrium.

We aggregate demand into macro-nodes, where commuters live at i and work at j. As common in the NDP, total demand n_{ij} between i and j is known. Commuters chose for their journey from i to j their transportation mode $m \in \{car, bus\}$ along a macro-route r through (several) sub-regions k. In this bathtub model [11], the macro-routes are not explicitly mapped to roads as only the macroscopic trip distance d_{ijmr} is important to obtain travel times T_{ijmr}. Using macro-routes does not require us to define a node-link model, but instead requires enumerating all macro-routes. Bus services on a macro-route operate with an average passenger route capacity of s_{ijr}.

Commuters choose mode and route based on the generalized cost of travel C that combine the monetary expenditures priced at rate p_m per unit trip distance and the time costs at rate π_m per unit of travel time and waiting time. We adopt a logit-based stochastic user equilibrium with scale parameter μ following the second Wardrop principle. Here, we assume that commuters choose mode and route with the lowest perceived costs resulting in passenger flows N_{ijmr}. The perceived costs are calculated with $\tilde{C}_{ijmr} = C_{ijmr} + 1/\mu \log(N_{ijmr})$.

The objective in the 3D-MFD-NDP as defined in Eqn. 1 is to minimize total travel time subject to constraints 2 and 3 for the network design variables and constraints 4-11 for the network flows and the user equilibrium. In general, all variables of the system must be non-negative. In Eqn. 2 we constrain the feasible set for the bus network following the bus network design approach by [9]. We further constrain the feasible set of network design parameters to the requirement that the entire multi-modal infrastructure must be self-funded, i.e. the monetary income must be equal to the infrastructure spending using Eqn. 3. For this, ω_m is the average price per lane-kilometer of infrastructure and σ are the average operational costs for a bus.

The set of possible speeds V_{mk} is constrained by the 3D-MFD. In particular, speeds depend on the accumulation of vehicles A_{mk} and network topology as indicated by Eqn. 4. The accumulation of cars is determined by Eqn. 5 with the share of macro-routes through a sub-region θ_{ijkmr}. With trip distance and speed, travel times T_{ijmr} are determined by Eqn. 6. The route costs C_{ijmr} combine time and monetary expenditures as given by Eqn. 7. Last, the sorting among routes and modes follows a logit based assignment as summarized in Eqn. 8.

Last, we define the user equilibrium as a mixed complementary problem (MCP) [12]. Route r is only chosen, i.e. $N_{ijmr} > 0$, if its costs are equal to the minimum cost $\tilde{C}_{ij} = \min_{mr}\left(\tilde{C}_{ijmr}\right)$ between i and j. If costs exceed $\tilde{C}_{ij}$, r is not used, i.e. $N_{ijmr} = 0$, as given by Eqn. 9. $\tilde{C}_{ij}$ is complementary to the node balance in Eqn. 10. We further define that in equilibrium, the flow of bus passengers is not exceeding the passenger capacity and if buses are used to capacity, passengers experience additional waiting time ρ_{ijr} as given by Eqn. 11. Therefore, we formulate the 3D-MFD-NDP as a mathematical program with equilibrium constraints (MPEC) [13].

$$\text{minimize} \quad \sum_{ijmr} N_{ijmr} T_{ijmr} \tag{1}$$

$$\text{subject to} \quad L_{k,\text{bus}} = \left(L_k/8 - l_k\right)^2 \left(1 + \alpha_k^2\right)/l_k \qquad 0 < \alpha_k \leq 1 \tag{2}$$

$$\sum_{km} \omega_m L_{km} + \sigma A_{k,\text{bus}} = \sum_{ijmr} p_m d_{ijmr} N_{ijmr} \tag{3}$$

$$\text{and} \quad V_{km} = \text{3D-MFD}_k\left(A_{km}; L_{km}; l_k; b_k; \Phi_k; \tau_k; \alpha_k\right) \tag{4}$$

$$A_{k,\text{car}} = \sum_{ijr} \theta_{ijk,\text{car},r} N_{ij,\text{car},r} \tag{5}$$

$$T_{ijmr} = \sum_{k} \theta_{ijkmr} \frac{d_{ijmr}}{V_{km}} \tag{6}$$

$$C_{ijmr} = p_m d_{ijmr} + \pi_m \left(T_{ijmr} + \rho_{ijr}\right) \tag{7}$$

$$N_{ijmr} = n_{ij} \frac{\exp\left(-\mu C_{ijmr}\right)}{\sum_{m'r'} \exp\left(-\mu C_{ijm'r'}\right)} \qquad \mu \geq 0 \tag{8}$$

$$\tilde{C}_{ijmr} - \tilde{C}_{ij} \geq 0 \qquad \perp N_{ijmr} \geq 0 \tag{9}$$

$$\sum_{mr} N_{ijmr} - n_{ij} = 0 \qquad \perp \tilde{C}_{ij} \geq 0 \tag{10}$$

$$s_{ijr} - N_{ij,\text{bus},r} \geq 0 \qquad \perp \rho_{ijr} \geq 0 \tag{11}$$

3 Discussion

The presented 3D-MFD-NPD is a novel approach to a widely discussed question of how to design cities for better or optimal transportation of passengers. Model extensions include reservoir dynamics and analyzing optimal pricing.

References

[1] Smeed, R. J., 1968. Traffic studies and urban congestion. Journal of Transport Economics and Policy 2, 33–70.

[2] Geroliminis, N., Daganzo, C. F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. Transportation Research Part B: Methodological 42, 759–770.

[3] Geroliminis, N., Zheng, N., Ampountolas, K., 2014. A three-dimensional macroscopic fundamental diagram for mixed bi-modal urban networks. Transportation Research Part C: Emerging Technologies 42, 168–181.

[4] Loder, A., Ambühl, L., Menendez, M., Axhausen, K. W., 2017. Empirics of multi-modal traffic networks using the 3D macroscopic fundamental diagram. Transportation Research Part C: Emerging Technologies 82, 88–101.

[5] Yang, H., H. Bell, M. G., 1998. Models and algorithms for road network design: a review and some new developments. Transport Reviews 18 (3), 257–278.

[6] Guihaire, V., Hao, J.-K., 2008. Transit network design and scheduling: A global review. Transportation Research Part A: Policy and Practice 42 (10), 1251–1273.

[7] Ben-Akiva, M. E., Lerman, S. R., 1985. Discrete choice analysis: theory and application to travel demand (Vol. 9): MIT press.

[8] Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. Transportation Research Part B: Methodological 46 (10), 1639–1656.

[9] Daganzo, C. F., 2010. Structure of competitive transit networks. Transportation Research Part B: Methodological 44 (4), 434–446.

[10] Loder, A., Bressan, L., Dakic, I., Ambühl, L., Bliemer, M., Menendez, M., Axhausen, K. W., 2019. Modeling multi-modal traffic in cities. Paper presented at the 98th Annual Meeting of the Transportation Research Board, Washington, D.C.

[11] Daganzo, C. F., 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. Transportation Research Part B: Methodological 41, 49–62.

[12] Nagurney, A., 2009. Network economics: a variational inequality approach. Advances in computational economics. Kluwer Academic Publishers.

[13] Luo, Z.-Q., Pang, J.-S., Ralph, D., 1996. Mathematical programs with equilibrium constraints. Cambridge: Cambridge University Press.

A Novel Traffic Estimation Approach Using Multi-Source Data on Motorways

Xuan Sy Trinh
Department of Civil and Natural Resources Engineering
University of Canterbury, New Zealand
Email: xuansy.trinh@pg.canterbury.ac.nz

Dong Ngoduy
Department of Civil and Natural Resources Engineering
University of Canterbury

Mehdi Keyvan-Ekbatani
Department of Civil and Natural Resources Engineering
University of Canterbury

Blair Robertson
School of Mathematics and Statistics
University of Canterbury

1 Introduction

Traditionally, traffic data are collected from fixed inductive loop detectors (LD), which are costly to be installed and maintained with high coverage in a large traffic network. Recently, thanks to the rapid growth in information technology, various technologies and systems can be used as additional sources of data, such as Bluetooth (BT) sensors, cellphone and GPS probes. These new data offer a potential opportunity to increase the penetration of detection and also extend the detection areas where installation is not economically feasible. Since each source of data carries only partial information about the traffic state and the measurements are often corrupted by noise, it is tempting to combine the information from different sources to increase accuracy, robustness and confidence in the estimation. However it is not an easy task due to three main issues: (i) heterogeneity in measurements from different sources; (ii) different spatio-temporal resolutions; and (iii) inconsistency in the multi-source measurements. Besides a large number of data-driven methods

in the literature, only a few model-based methods have been recently investigated to fuse data from heterogeneous sources. A modified version of the standard extended Kalman Filter (EKF), called the Incremental EKF, has been used to integrate observations from loop detectors with partial observations from Bluetooth and GPS devices [1]. A Progressive Extended Kalman Filter (PEKF) method has also been used to combine measurements from the wireless communication records and microwave sensors [2]. However, both of these methods were developed based on the EKF, which may suffer from two important drawbacks when applied in real-world problems: the performance is poor when the system is severely nonlinear and/or multi-modal, dynamic model and measurement model functions must be differentiable, and Jacobian matrices can be difficult to calculate and prone to errors. To address these issues the Unscented Kalman Filter (UKF), a superior alternative to the EKF in many applications, is used in this research.

In addition to UKF, the Unscented Information Filter (UIF) is also considered in this work due to the simplicity of the update step, which makes it suitable for multiple sensor estimation. In general, the UIF and UKF are algebraically equivalent and can produce the same estimates and the same estimation error covariances. The only difference is the UIF works with information matrices and information vectors instead of the predicted covariance matrices and predicted states. The fusion procedure in the UIF is remarkably simpler than in the UKF, especially when the number of measurement is significantly larger than the size of the state space. This is because the UIF update equations are additive, which is capable of integrating measurements by simply adding their information to the information vector and information state. To the best of our knowledge, the UIF has not been used with data fusion problems in traffic. Therefore, the work here considers applying the UIF in the context of traffic state estimation from multiple data sources and a comparison with the widely known UKF is made.

In order to achieve good performance, both the UKF and the UIF require an appropriate selection of measurement noise covariances. In many works, these covariances are chosen based on experience or trial and error, and they remain unchanged during the filer process. This practice could be problematic especially when the covariances suddenly change or when the measurement noise is greatly influenced by the working environment of the sensors. One of the efficient ways to overcome this problem is to use an adaptive algorithm. In this paper we consider the adaptive UKF, which adaptively update the measurement noise covariances with the information obtained during the filter process.

2 Methodology

Recently [1] and [2] proposed methods to fuse data from multiple heterogeneous sources based on the extended Kalman Filter (EKF). Unfortunately, the EKF has two major drawbacks. First,

the first-order expansion used by the EKF algorithm is a poor approximation for most non-linear functions. This can lead to large errors in the estimates and sometimes divergence of the filter, especially when the time step is not sufficiently small. Second, the EKF requires the calculation of the Jacobian matrices, which can be troublesome as the model function and the sensor function might not be continuous across the range in which the linearization is used. To address these limitations, the UKF can be used in the place of the EKF [3], [4]. In the UKF, the mean and variance of the true state are represented by a set of sample points known as *sigma points*, which are selected deterministically through a process named the Unscented Transform. When these points propagate through the non-linear functions, they can capture the posterior mean and covariance without linearizing using Jacobian matrices. Hence, the UKF is is a powerful alternative to the EKF in traffic data fusion problems. In this work, we combine the UKF with the incremental algorithm in [1] to integrate different data sources incrementally with no state transition in between. This method is named as the incremental UKF.

The Unscented Information Filter is derived from the Unscented Kalman Filter, in which the information state and information matrix are used instead of the state and covariance. They are defined as

$$\mathbf{Y} = \mathbf{P}^{-1}, \quad \hat{\mathbf{y}} = \mathbf{P}^{-1}\hat{\mathbf{x}} \tag{1}$$

where $\hat{\mathbf{x}}$ and $\mathbf{P}$ are the estimated state and its covariance, $\hat{\mathbf{y}}$ and $\mathbf{Y}$ are the information state and the information matrix. Using this approach, the update equations are:

$$\mathbf{Y}_{t|t} = \mathbf{Y}_{t|t-1} + \sum_{o} \mathbf{I}_{t,o}, \quad \hat{\mathbf{y}}_{t|t} = \hat{\mathbf{y}}_{t|t-1} + \sum_{o} \mathbf{i}_{t,o} \tag{2}$$

where $\mathbf{i}_{t,o}$ and $\mathbf{I}_{t,o}$ are the information state contribution and the associated information matrix from each measurement. As can be seen from (2), the measurement update equations are additive, which makes the UIF suitable for data fusion by nature. The fusion process in the UIF is therefore much simpler than that in the UKF, especially when the number of measurements is relatively large.

In this paper, the adaptive algorithm for the UKF is adopted from [6]. However, we only consider the adaptation of measurement noise instead of both measurement noise and system noise because of the inaccuracy and inconsistency of system noise information estimated after each data souce is fused. At each time step, a faulty detection mechanism is used to assess the accuracy of the current measurement noise covariances. If the fault is detected, these covariances will be adjusted based on the weighted sum of the previous estimation and the current theoretical one as follows:

$$R_t = (1-\sigma)R_{t-1} + \sigma[\epsilon_t \epsilon_t^T + S_{t|t}^{zz}] \tag{3}$$

where R_t is the measurement noise covariance at time t, σ is a weighting factor to be chosen, ϵ_t is the residual vector, and $S^{zz}_{t|t}$ is the estimated innovation covariance matrix.

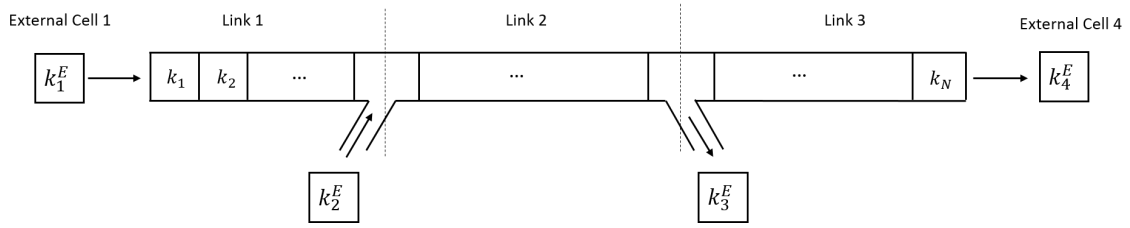


Figure 1: A toy traffic network

A toy traffic network to test the methods has the structure similar to the one shown in the Fig. 1. Using this layout the synthetic data were obtained from a microscopic simulator, namely AIMSUN. While the inductive loop traffic detectors in the network were set up to detect all the passing vehicles, the GPS probes and Bluetooth scanners were configured in the way that only a fraction of traffic can be detected. This is to replicate practical situations where not every vehicle carries a GPS or Bluetooth device. The penetration rates for GPS and Bluetooth are 2% and 20% respectively. Due to the low penetration rates, GPS or Bluetooth observations could be missing if there is no GPS or Bluetooth vehicle detected during the sample times. Here we assume that the internal traffic information is provided by only GPS and Bluetooth sensors with low penetration rates, and the external traffic information is solely obtained by loop detectors. For all the reasons above, recovering the true states in this case could be challenging and may be less accurate than using loop detectors alone for the whole network.

3 Results and Conclusions

The experimental results show that both the UKF and the UIF are capable of fusing data from multiple sources with comparable accuracy, even if the penetration rates are low. The UKF appears to perform better than the UIF when the noise covariance matrices are selected appropriately. In contrast, the UIF has better performance when those noise covariances are chosen incorrectly. The results also show that the Bluetooth data contribute insignificant part in improving the accuracy of the estimations.

In the case of incorrect noise covariances, by applying the adaptive algorithm, the UKF outperforms the UIF alone, and provides significantly better results than the standard UKF. However, when the noise covariances have been chosen properly, the adaptive UKF may be slightly less accurate than the standard UKF in estimation.

References

[1] Nantes, Alfredo, et al. "Real-time traffic state estimation in urban corridors from heterogeneous data." *Transportation Research Part C: Emerging Technologies* 66 (2016): 99-118.

[2] Liu, Yingshun, et al. "A Progressive Extended Kalman Filter Method for Freeway Traffic State Estimation Integrating Multisource Data." *Wireless Communications and Mobile Computing* 2018 (2018).

[3] Julier, Simon, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte. "A new method for the nonlinear transformation of means and covariances in filters and estimators." *IEEE Transactions on automatic control* 45.3 (2000): 477-482.

[4] Julier, Simon J., Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. "A new approach for filtering nonlinear systems." *American Control Conference, Proceedings of the 1995*. Vol. 3. IEEE, 1995.

[5] Wan, Eric A., and Rudolph Van Der Merwe. "The unscented Kalman filter for nonlinear estimation." *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000.

[6] Zheng, Binqi, et al. "A robust adaptive unscented Kalman filter for nonlinear estimation with uncertain noise covariance." *Sensors* 18.3 (2018): 808.

Hyperconnected Urban Parcel Logistic Systems Design

Benoit Montreuil, Sara Kaboudvand, Louis Faugere, Martin Savelsbergh

Physical internet Center, Supply Chain & Logistics Institute

School of Industrial & Systems Engineering

Georgia Institute of Technology

Corresponding Author: Benoit Montreuil

School of Industrial & Systems Engineering

Georgia Institute of Technology, Atlanta, U.S.A.

Email: benoit.montreuil@isye.gatech.edu

Extended Abstract

Ever more of the world population of people, businesses and institutions is inhabiting urban agglomerations. Fulfilling demand for products and services has become a critical challenge in most cities, metropolises and megacities across the world. From a business perspective, this reflects in the emergence of fast, precise, seamless, reliable, efficient and sustainable omnichannel supply chains, of last-mile delivery and reverse logistics as top competitiveness challenges. From an urban authority perspective, this reflects in smart city logistics initiatives and policies aiming for this urban demand fulfillment to be done in harmony with the goals toward socioeconomical development, quality of citizen life and overall sustainability, notably trying to minimize traffic and congestion, exploiting smartly and harmonizing beautifully with existing urban infrastructures and assets [1], [2].

It is thus not surprising that parcel logistics, driven by both business-to-consumer and business-to-business demand for fast pickup and delivery of small orders, is undergoing huge transformation all around the world, and especially in megacities. In the past, parcel logistic systems were designed for insuring delivery of picked up parcels within a few days, at the fastest during the next day. Currently, same-day and next-day delivery is the typical norm. Already the future trend is clearly emerging: the competitive edge is toward efficiently achieving on a large scale the delivery within a few hours, even a few minutes, depending on the time of day and location of the pickup and delivery locations, or yet reliably within a short time window specified by the client.

This paper addresses the design foundations of hyperconnected urban parcel logistic systems capable of meeting the challenges expressed above. It builds on generic Physical Internet foundations, as introduced by [3] and [4]. The Physical Internet is a hyperconnected global logistic system enabling seamless open asset sharing and flow consolidation through standardized encapsulation, modularization, protocols and interfaces to improve the capability, the efficiency and the sustainability of fulfilling humanity's demand for the services of physical objects. It is said to be hyperconnected as its components and actors are intensely interconnected on multiple layers, ultimately anytime, anywhere. Interconnectivity layers notably include digital, physical, operational, business, legal and personal

interconnectivity. Key building blocks of the Physical Internet include a unified set of standard modular logistic containers [5]; modular-container centric logistics equipment and technology; standard logistics protocols; certified open logistic facilities and ways; global logistic monitoring system; open logistic decision and transaction platforms; smart data-driven analytics, optimization and simulation tools; and certified open logistic service providers.

The paper more specifically builds upon two threads of research and innovation: hyperconnected city logistics [2] and hyperconnected omnichannel supply chains and logistic systems [6].

As introduced in [2], hyperconnected City Logistics fuses the foundations of City Logistics and the Physical Internet. It has been introduced through nine key concepts: (1) Interconnect cities as nodes of the world's logistic web; (2) Interconnect cities by systems standardization; (3) Interconnect the multi-faceted activities of city logistics (including multimodal transportation, crossdocking, transshipment and handling activities, as well as supply, value-adding, storage and deployment activities); (4) Interconnect city logistics networks in an urban web architecture; (5) Interconnect the multiplicity of urban logistic centers; (6) Interconnect city logistics stakeholders into an open system; (7) Interconnect goods through modular logistic containers; (8) Interconnect People Mobility and Freight Logistics in the city; and (9) Interconnect city logistics with urban planning. As described in [6], hyperconnected omnichannel supply chains and logistic systems exploit Physical Internet foundations for efficiently and sustainably enabling at large scale customers to purchase their goods using any channel (e-commerce site, mobile app, retail store, etc.) and to get them according to their choice such as ship-to-home, ship-to-me, pick-at-locker, pick-at-drive and pick-at-store. To achieve this without excessive waste of resources and duplication of assets while being highly responsive, they are based on the seamless interconnection of omnichannel product transportation; goods deployment, pickup and delivery; and goods production.

Beyond the direct application of concepts from the above threads (such as using standard modular containers for example), this paper exposes the foundations for designing and operating hyperconnected urban parcel logistic systems according to four key characteristics.

First, hyperconnected urban parcel logistic systems exploit a multi-tier space pixelization, as illustrated in Figure 1. At lowest tier lies unit zones corresponding to blocks, large buildings, campuses, etc. At the second tier, local cells cluster contiguous unit zones. At the third tier, urban areas cluster contiguous local cells. At the fourth, fifth and sixth tiers respectively , regions cluster urban and non-urban areas, blocks cluster regions, and the world clusters blocks.

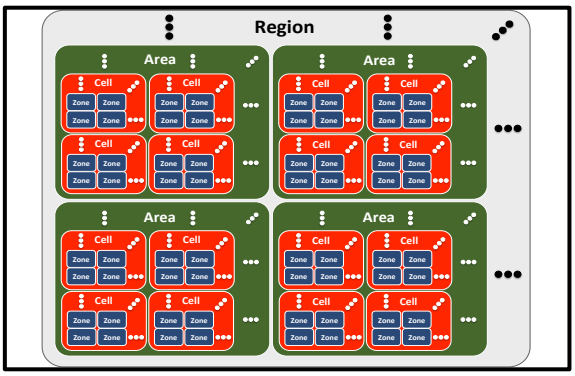


Figure 1. Multi-Tier Pixelization of Urban Space

Second, hyperconnected urban parcel logistic systems are based on the exploitation and interconnection of multi-party open logistic hubs, adapted to each spatial tier of pixelization. For example, as depicted in Figure 2, access hubs are located at the intersection of access zones, local hubs at the intersections of local cells, and gateway hubs at the intersections of urban areas.

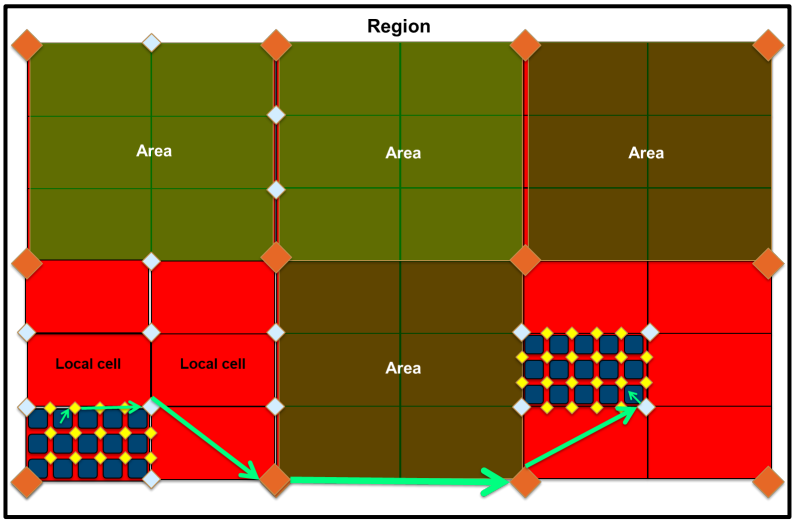


Figure 2. Access, Local and Gateway Hubs Interconnecting the Urban Space

Third, they are interconnecting these logistic hubs, creating an urban logistic web composed of interconnected multi-plane meshed networks, as illustrated in Figure 3. This allows, as exemplified in Figure 2, to have a parcel picked up in a zone, brought to one of its access hubs, then to a nearby local hub to be then moved to a gateway hub to be directed to a gateway hub from another part of the city, then be moved to a local hub and then the access hub most convenient for delivering to the customer place or a nearby smart locker, as selected by the client. Contrary to a typical hub-and-spoke network, here at each plane there is a meshed network, and parcels are not forced extensive extra travel when moving from any part or any other part of a city. Intercity parcels get into the city and out of the city through gateway hubs, interconnected through regional and global hubs as pertinent (see Figure 4).

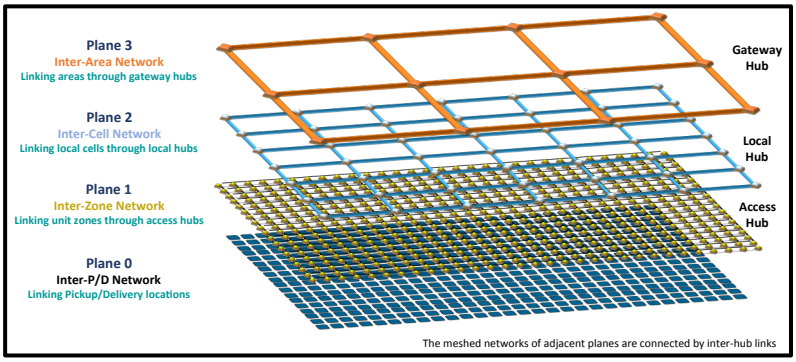


Figure 3. Urban Logistic Web Composed of Interconnected Multi-Plane Meshed Networks

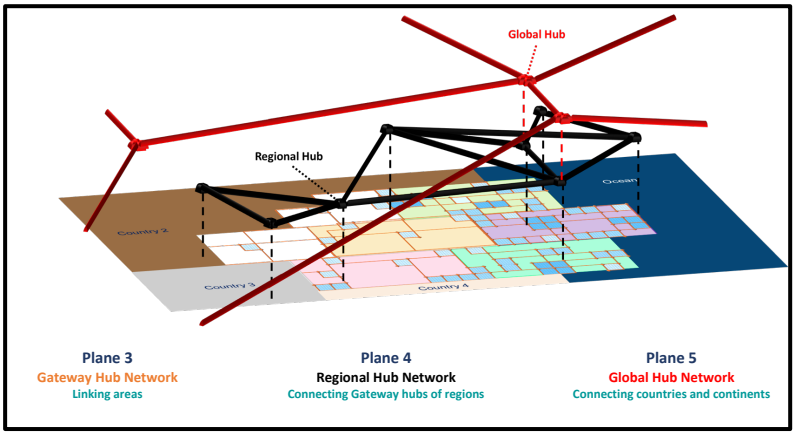


Figure 4. Interconnecting the Inter-Area Network with Inter-Region and Inter-Block Networks

Fourth, hyperconnected urban parcel logistic systems rely on data-centric, distributed yet interconnected decision making for dynamic and concurrent (1) parcel routing, (2) parcel consolidation, (3) vehicle routing, (4) service offer design, so as to serve as best as possible each client's demand for parcel pickup and delivery, achieving high service levels and high multi-party asset utilization. Such decision making is based on the smart application of optimization models and heuristics, and of machine learning algorithms. The paper describes in further detail a potential decision architecture.

Finally, the paper presents results from a simulation-based experiment assessing the implementation of a hyperconnected urban parcel logistic system in Shenzhen, China, a megacity with frequently a million pickup/delivery transaction a day into, out of and within its boundaries. Figure 5 displays a simulation screen in action. Key performance indicators are provided to contrast expected performance versus baseline contemporary hub-and-spoke parcel logistic systems.

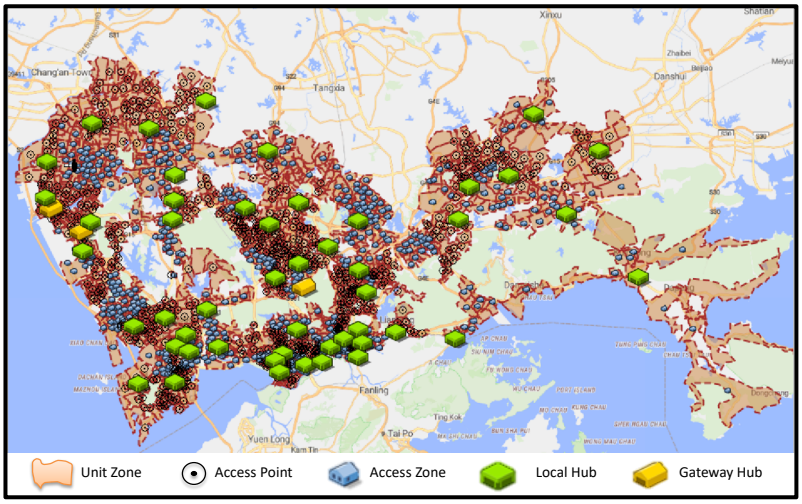


Figure 5. Experimenting in Shenzhen: a Million+ Parcels a Day Into, Within and Out of the Megacity

References

[1] M. Savelsbergh. and T. V. Woensel, "City Logistics: Challenges and Opportunities," Transportation Science, vol. 50, no. 2, pp. 579-590, 2016.

[2] T. G. Crainic & B. Montreuil, "Physical Internet Enabled Hyperconnected City Logistics", in *Transportation Research Procedia – Tenth International Conference on City Logistics*, v12, 383-398, 2016.

[3] B. Montreuil, "Toward a Physical Internet: meeting the global logistics sustainability grand challenge," Logistics Research, vol. 3, no. 2-3, pp. 71-87, 2011.

[4] B. Montreuil, R.D. Meller & E. Ballot, "Physical Internet Foundations", in *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, ed. T. Borangiu, A. Thomas and D. Trentesaux, Springer, p. 151-166, 2013.

[5] B. Montreuil, E. Ballot and W. Tremblay, "Modular Design of Physical Internet Transport, Handling and Packaging Containers," in Progress in Material Handling Research, Vol. 13, Charlotte, NC, U.S.A., 29 p., 2016.

[6] B. Montreuil, "Omnichannel Business-to-Consumer Logistics and Supply Chains: Towards Hyperconnected Networks and Facilities", *Progress in Material Handling Research Vol. 14*, Ed. K. Ellis et al., MHI, Charlotte, NC, USA, 32 p., 2017.

[7] B. Montreuil, S. Buckley, L. Faugere, K. Reem, S. Derhami, "*Urban Parcel Logistic Hub and Network Design: The Impact of Modularity and Hyperconnectivity*", in Progress in Material Handling Research, Vol. 15, 8 p., 2018.

Integrated Scheduling and Flow Management in Air Traffic Management Networks

Kai Wang

Department of Logistics and Maritime Studies, Hong Kong Polytechnic University

Alexandre Jacquillat

Heinz College, Carnegie Mellon University, ajacquil@andrew.cmu.edu

Air traffic management systems have been facing strong demand growth, while available infrastructure has remained limited. The resulting imbalances between demand and capacity can lead to severe congestion and/or unmet demand at busy airports and at busy times. In the United States, the costs induced by air traffic congestion were estimated at $31.2 billion in 2007, borne by airlines, air travelers and society [Ball et al., 2010]. Absent capacity expansion opportunities, the two major congestion mitigation levers are *scheduling interventions* and *air traffic flow management*.

Scheduling interventions refer to demand management rules governing flight scheduling. These are *strategic* measures, implemented months in advance of operations. Outside the U.S., busy airports are subject to "schedule coordination": they declare a value of capacity (usually set close to the airport's capacity under poor weather conditions), and allocate slots accordingly to the airlines. Demand management is much more limited at U.S. airports. By and large, flight schedules are not constrained by any declared capacity. Only a few of the busiest airports are subject to "flight caps", which are much less stringent than declared capacities in place at comparable schedule-coordinated airports. These variations highlight a trade-off between schedule coordination focused on congestion mitigation and *laissez-faire* focused on high scheduling levels [de Neufville and Odoni, 2013].

Air traffic flow management (ATFM) refers to operating procedures aimed to optimize aircraft flows across networks of airports and air traffic control sectors. It is a *tactical* measure, implemented during each day of operations. Overall, ATFM aims to control the departure and arrival times of the flights and the speed, routing and altitude of en-route aircraft in order to absorb flight delays at departure airports or in the en-route airspace rather than in the terminal airspace, where their costs are higher. These initiatives have been successfully implemented in the US and in Europe.

The strategic SI problem and the tactical ATFM problem are interdependent. Indeed, the optimal aircraft flows depend on the schedule of flights. Vice versa, the optimal schedule of flights depends on the tactical capabilities of ATFM. All else equal, the more effectively ATFM can miti-

gate delay costs, the less aggressive SI needs to be to guarantee any targeted level of service. These interdependencies have been addressed at a single airport [Jacquillat and Odoni, 2015]; however, the joint network-wide dynamics of SI and ATFM remain unexplored. These interdependencies are complicated by the uncertainty at the time of scheduling regarding the operating conditions that will prevail at the time of operations. Thus, one cannot simply set a schedule that will match capacity with certainty across spatial-temporal networks. This creates a trade-off between absorbing the costs of demand-capacity imbalances through demand management vs. flight delays.

We propose an original integrated approach to air traffic management that jointly optimizes strategic SI decisions and tactical ATFM decisions in capacity-constrained networks, under operating uncertainty. Specifically, it optimizes flight schedules in a network of airports, while capturing how ATFM systems will respond to any schedule of flights in various operating scenarios. This contrasts with the SI literature, which does not consider the effect of flight schedules on operations, and with the ATFM literature, which considers a fixed schedule of flights as an input. Methodologically, this is formulated as a two-stage stochastic integer program, and relies on a new solution method with provable quality guarantees to address this class of problems.

Integrated Network-wide Scheduling and Flow Management Model

We formulate a novel *Integrated Network-wide Scheduling and Flow Management Model* (INSFMM) as a two-stage stochastic integer program. The model takes as inputs the airlines' preferred schedule of flights and the operating capacity of each airport and air traffic controls sector in the network. The first stage corresponds to scheduling interventions (SI). It determines flight schedules to minimize the schedule displacement (i.e., the deviations from airlines' requests), subject to scheduling and network connectivity constraints. These scheduling decisions are made before operating uncertainty is resolved, thus relying on probabilistic characterizations of air traffic operations. The second stage corresponds to ATFM. It optimizes flight operations to minimize delays from flights' scheduled times, subject to flight operating and capacity constraints at each airport and air traffic control sector. These operating decisions are made as information on weather and other operating conditions becomes available. The model is formulated as a bi-objective problem that trades off schedule displacement and expected flight delays.

Specifically, the (INSFMM) is formulated as follows, where ε is a weight parameter:

$$\begin{aligned} \text{(INSFMM)} \qquad \min \quad & \varepsilon \cdot \text{Schedule displacement} + (1-\varepsilon) \cdot \mathbb{E}(\Psi) \\ \text{s.t.} \quad & \text{Flight scheduling constraints} \\ & \text{Network connectivity constraints,} \end{aligned}$$

where the second-stage objective Ψ is given by:

$$\begin{aligned} \Psi = \min \quad & \text{Flight delay} \\ \text{s.t.} \quad & \text{Flight operating constraints} \end{aligned}$$

Network connectivity constraints
Airport/sector capacity constraints

We propose valid constraints that tighten the linear programming relaxation of (INSFMM). Nonetheless, both SI and ATFM involve large-scale and complex integer optimization problems, so the (INSFMM) is highly intractable in realistic test instances using commercial solvers.

Two-stage Stochastic Integer Programming Solution Approach

Much progress has been made to solve stochastic continuous programs by means of Benders decomposition [Birge and Louveaux, 2011]. Benders decomposition iterates between a master problem, which provides a feasible first-stage solution, and sub-problems, which use the recourse function to generate valid cuts into the master problem. However, this is not directly applicable in stochastic integer programming because the recourse function is non-convex. A standard stochastic integer programming approach is the integer L-shaped method, which also decomposes the problem into a master problem and sub-problems and uses valid cuts based on the second-stage objective function value [Laporte and Louveaux, 1993]. This method has been applied in various domains, but (INSFMM) remains orders of magnitude larger than what has been solved thus far.

To address this challenge, we propose a new solution approach with provable quality guarantees for two-stage stochastic integer programming—illustrated in Figure 1. Like Benders decomposition, the approach decomposes (INSFMM) into a master problem and scenario-specific sub-problems. However, instead of solving the second-stage problem as an integer program, we derive new cuts based on its dual linear programming (LP) relaxation. These cuts stem from a new theoretical result that provides a lower bound of the second-stage objective function, expressed as a function of the first-stage decision variables and the reduced cost of the dual LP sub-problem relaxation.

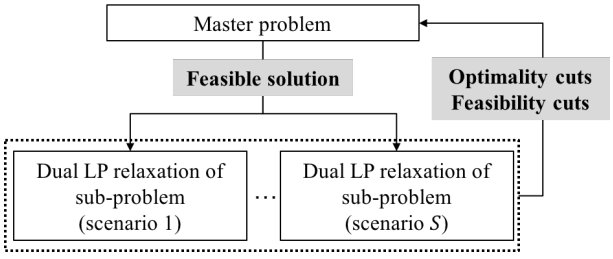


Figure 1: Overview of proposed two-stage stochastic integer programming solution approach

We use this approach to develop a solution algorithm that incorporates original neighborhood constraints and applies acceleration techniques based on local branching and Pareto-optimality cuts. We also propose a novel scenario reduction method to generate representative scenarios in stochastic programs. Results show that this algorithm yields near-optimal solutions to (INSFMM) for networks of the size of the US National Airspace System in reasonable computational times.

In summary, the proposed method provides (i) a new solution approach for two-stage stochastic integer programming that leverages the dual LP relaxation of the second-stage problem, (ii) new optimality cuts in Benders decomposition algorithms, and (iii) high-quality solutions to the largest stochastic integer programs implemented thus far in the literature.

Computational Results and Implications

From a practical standpoint, (INSFMM) provides decision-making support to optimize scheduling interventions across a network of airports, while accounting for ATFM dynamics. This contributes to the literature on demand management by optimizing scheduling interventions across multiple airports in a network. More importantly, it provides a novel conceptual approach that balances the strategic costs of scheduling interventions and the tactical costs of flight delays. At schedule-coordinated airports outside the U.S., this can support the setting of declared capacities and slot allocation at multiple airports simultaneously. This would augment existing decentralized practices where (i) capacity declaration is left up to each airport, leading to a lack of standardization, and (ii) slot allocation is performed at each airport independently and network-wide conflicts are then resolved in an *ad hoc* manner at bi-annual slot conferences, leading to potentially sub-optimal decisions at the network level. At U.S. airports, the proposed approach provides objective and transparent decision-making support to inform the appropriate extent of scheduling interventions. Moreover, computational results inform *where* (i.e., at which airport) and *when* (i.e., at which times of the day) scheduling adjustments can be most effective. This contrasts with existing approaches where "flight caps" are applied at a few airports and does not vary by time of day. Ultimately, the approach developed here can support network-wide scheduling practices to maximize scheduling benefits and minimize congestion costs for airlines, passengers and other stakeholders.

References

[Ball et al., 2010] Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., and Zou, B. (2010). Total Delay Impact Study. Technical report, National Center of Excellence for Aviation Operations Research, College Park, MD.

[Birge and Louveaux, 2011] Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

[de Neufville and Odoni, 2013] de Neufville, R. and Odoni, A. (2013). *Airport Systems: Planning, Design and Management*. McGraw-Hill, 2nd edition.

[Jacquillat and Odoni, 2015] Jacquillat, A. and Odoni, A. (2015). An Integrated Scheduling and Operations Approach to Airport Congestion Mitigation. *Operations Research*, 63(6):1390–1410.

[Laporte and Louveaux, 1993] Laporte, G. and Louveaux, F. (1993). The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133–142.

Simultaneous correction of the time and location bias associated with a reported crash by exploiting the spatiotemporal evolution of travel speed

Zhengli Wang
Department of Industrial Engineering
Tsinghua University

Hai Jiang
Department of Industrial Engineering
Tsinghua University
Email: haijiang@tsinghua.edu.cu

1 Introduction

Accurate occurrence time and location of a reported crash are critical to effective crash analysis. However, it has been widely recognized in the literature that the occurrence times and locations recorded in crash reports are often biased [1, 2], that is, they are often different from the actual occurrence time and location of the crash. The time bias is primarily caused by the fact that many crash reporting systems automatically log the time the crash is reported to authorities, for example, the time the phone call is received, and store this automatically logged time as the occurrence time. The location bias is often caused by the lack of standards in the textual description of the crash location [3].

Although there has been a proliferation of studies that attempt to correct the bias associated with a reported crash, most, if not all, of them focus exclusively on correcting the bias in location. Early studies rely on the distances between the reported crash location and adjacent roads to correct the location bias. For example, a buffer zone is created with a predefined radius to identify the matching link in [4], while the crash location is adjusted to the closest junction in [5]. Later, road name filtering is incorporated in this process in [6]. When the driving directions of the vehicles are available, more sophisticated approaches are developed. For example, a weighted score scheme that combines the perpendicular distance of the reported crash location to each candidate link and the angular difference between the driving directions and the link directions to correct the location bias is used in [1]. Recently, artificial neural network that considers road name, road type, direction of travel, and recorded crash location to determine the link on which the crash occurred is used in [7]. This approach is later extended by [8], in which fuzzy logic is applied to identify candidate links, and by [9], in which the multilevel logistic regression model is adopted with the distance and direction differences as explanatory variables.

In this research, we propose to simultaneously correct the time and location bias associated with a reported crash, which is new to the literature. For a given crash, we first follow the procedure detailed in [8] to identify the set of candidate links in the vicinity of the reported crash site. We then examine the spatiotemporal evolution of travel speed on these candidate links and select the one that is most congruent with the occurrence of a crash. We formulate the candidate selection process as an integer programming model and develop a set of novel constraints to estimate the spatiotemporal impact regions, which characterize the evolution of speed in the speed contour plots of the candidate links. We finally use the time and location when travel speed begins to drop to correct the time and location bias associated with the crash. We validate our model using real crash data in Beijing and find that our model can reduce the average bias in time from 7.8 minutes to 1.7 minutes, or a 78.21% reduction; and reduce the average bias in location from 0.140 kilometer to 0.025 kilometer, or a 82.14% reduction.

2 The Modeling Framework

For a given crash, we first get its coordinates, that is, its longitude and latitude, by the textual description of its location using geocoding functions provided by Google Map [10, 11]. We then identify candidate links following the method detailed in [8]: An error circle is drawn around the reported crash location and road links that fall within or intersect with the error circle are considered to be candidate links. We then construct the speed contour plot, by which the evolution of travel speed on a link can be conveniently visualized. To construct the speed contour plot of a link, we first discretize time into equal intervals indexed by m. We choose the analysis period so that it starts at an interval prior to the reported occurrence time and ends at an interval when the impact of the crash completely dissipates. We number the intervals in the analysis period from 1 to M and let m^* be the interval that corresponds to the reported occurrence time of the crash. Assuming there are altogether N candidate links indexed by n, we discretize these links into sections of equal lengths. For link n, let J_n be the total number of sections and these sections are numbered from 1 to J_n from the upstream to the downstream.

For link n, let $s_{j,m,n}$ be the travel speed on section j in time interval m. Suppose that a crash took place on link n in a given day. We can get the *crash-induced* values of $s_{j,m,n}$, denoted as $\hat{s}_{j,m,n}$, and we can produce the *crash-induced* speed matrix. Using historical observations of $s_{j,m,n}$ during days when there were no crashes, we can obtain the *crash-free* mean and standard deviation of $s_{j,m,n}$, denoted as $(\bar{s}_{j,m,n}, \sigma_{j,m,n})$. If $\hat{s}_{j,m,n}$ is significantly smaller than $\bar{s}_{j,m,n}$, for example, if $\hat{s}_{j,m,n} \leq \bar{s}_{j,m,n} - \alpha\sigma_{j,m,n}$, where α is a positive threshold parameter trained using existing data [12, 13], we say that cell $\langle j, m\rangle$ is impacted by the crash on that given day. Let us use discriminant binary indicator $P_{j,m,n}$ to indicate whether $\hat{s}_{j,m,n}$ is significantly lower than $\bar{s}_{j,m,n}$, that is, if $\hat{s}_{j,m,n} \leq \bar{s}_{j,m,n} - \alpha\sigma_{j,m,n}$, $P_{j,m,n} = 1$; otherwise $P_{j,m,n} = 0$.

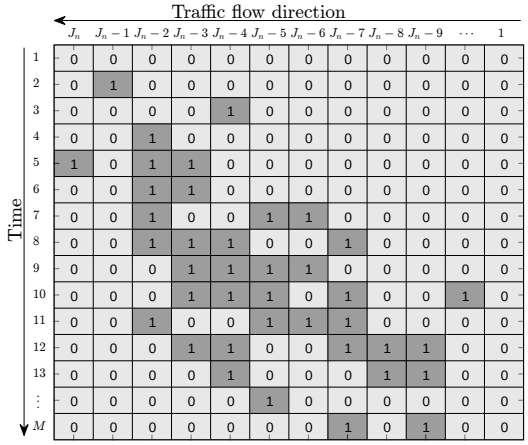


Figure 1: Illustration of binary indicators $P_{j,m,n}$, which depends on $\bar{s}_{j,m,n}$ and $\hat{s}_{j,m,n}$.

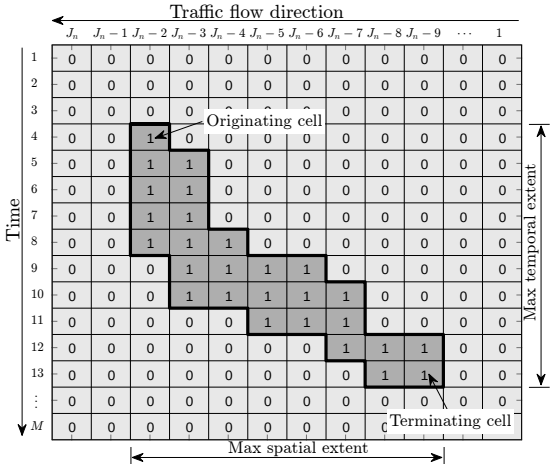


Figure 2: The desired shape of the impact region for Figure 1.

In Figure 1, we give an example for the values of $P_{j,m,n}$ and color cells with $P_{j,m,n} = 1$ in dark gray. According to [13] and [14], the shape of the impact region formed by cells in dark gray should conform to the propagation of shockwaves, while the impact region in this figure does not. This is not unusual due to the presence of noise and stochasticity in the speed data reported by probe vehicles [13, 14]. In Figure 2, we show the desired shape of the impact region and the desired values of $P_{j,m,n}$. It is not difficult to verify that its shape conforms to the propagation of shockwaves. To recover the desired shape of the impact region, we introduce binary decision variables for each cell: (1) If cell $\langle j, m\rangle$ on link n is indeed impacted by the crash, $\delta_{j,m,n} = 1$; otherwise $\delta_{j,m,n} = 0$; (2) If the crash originates from cell $\langle j, m\rangle$ on link n, $\gamma_{j,m,n} = 1$; otherwise $\gamma_{j,m,n} = 0$; and (3) If the crash terminates at cell $\langle j, m\rangle$ on link n, $\zeta_{j,m,n} = 1$; otherwise $\zeta_{j,m,n} = 0$.

Once we know the values of $\gamma_{j,m,n}$, we can get the estimated occurrence time (denoted as $\hat{m}$), link ID (denoted as $\hat{n}$), and section (denoted as $\hat{j}$): $\hat{m} = \sum_{j=1}^{J_n}\sum_{m=1}^{M} m \cdot \gamma_{j,m,n}$, $\hat{n} = \sum_{n=1}^{N} n \cdot \left(\sum_{j=1}^{J_n}\sum_{m=1}^{M}\gamma_{j,m,n}\right)$ and $\hat{j} =$

$\sum_{j=1}^{J_n} \sum_{m=1}^{M} j \cdot \gamma_{j,m,n}$.

We develop the following optimization model to find the values of $\delta_{j,m,n}$, $\gamma_{j,m,n}$, and $\zeta_{j,m,n}$, so that they not only conform to the propagation of shockwaves, but also have minimal deviation from $P_{j,m,n}$:

$$\text{minimize} \quad \sum_{n=1}^{N} \sum_{j=1}^{J_n} \sum_{m=1}^{M} \left[P_{j,m,n} \cdot (1 - \delta_{j,m,n}) + (1 - P_{j,m,n}) \cdot \delta_{j,m,n} \right].$$

subject to

$$\sum_{n=1}^{N} \sum_{j=1}^{J_n} \sum_{m=1}^{M} \gamma_{j,m,n} = 1; \tag{1}$$

$$\sum_{n=1}^{N} \sum_{j=1}^{J_n} \sum_{m=1}^{M} \zeta_{j,m,n} = 1; \tag{2}$$

$$\gamma_{j,m,n} \le \delta_{j,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{3}$$

$$1 - \gamma_{j,m,n} \ge \delta_{j+1,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{4}$$

$$1 - \gamma_{j,m,n} \ge \delta_{j,m-1,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{5}$$

$$\delta_{j,m-1,n} + \delta_{j+1,m,n} \ge \delta_{j,m,n} - \gamma_{j,m,n}, \qquad \forall 1 \le n \le N, 1 \le j \le J_n, 1 \le m \le M; \tag{6}$$

$$\sum_{1 \le j \le J_n} \sum_{1 \le m \le M} \delta_{j,m,n} \le M J_n \sum_{1 \le j \le J_n} \sum_{1 \le m \le M} \gamma_{j,m,n}, \qquad \forall 1 \le n \le N; \tag{7}$$

$$\zeta_{j,m,n} \le \delta_{j,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{8}$$

$$1 - \zeta_{j,m,n} \ge \delta_{j-1,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{9}$$

$$1 - \zeta_{j,m,n} \ge \delta_{j,m+1,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{10}$$

$$\delta_{j,m+1,n} + \delta_{j-1,m,n} \ge \delta_{j,m,n} - \zeta_{j,m,n}, \qquad \forall 1 \le n \le N, 1 \le j \le J_n, 1 \le m \le M; \tag{11}$$

$$\sum_{1 \le j \le J_n} \sum_{1 \le m \le M} \delta_{j,m,n} \le M J_n \sum_{1 \le j \le J_n} \sum_{1 \le m \le M} \zeta_{j,m,n}, \qquad \forall 1 \le n \le N; \tag{12}$$

$$\sum_{n=1}^{N} \sum_{j=1}^{J_n} \left(\sum_{m=1}^{M} m \cdot \zeta_{j,m,n} - \sum_{m=1}^{M} m \cdot \gamma_{j,m,n} \right) \ge \Theta; \tag{13}$$

$$\sum_{n=1}^{N} \sum_{m=1}^{M} \left(\sum_{j=1}^{J_n} j \cdot \gamma_{j,m,n} - \sum_{j=1}^{J_n} j \cdot \zeta_{j,m,n} \right) \ge \Phi; \tag{14}$$

$$\sum_{n=1}^{N} \sum_{j=j_n^* - \Delta_n^-}^{j_n^* + \Delta_n^+} \sum_{m=m^* - \Lambda^-}^{m^* + \Lambda^+} \gamma_{j,m,n} = 1; \tag{15}$$

$$\delta_{j+1,m,n} + \delta_{j,m-1,n} - 1 \le \delta_{j,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{16}$$

$$\delta_{j+1,m,n} + \delta_{j,m-1,n} \ge \delta_{j,m,n} - \gamma_{j,m,n}, \quad \forall 1 \le j \le J_n, 1 \le m \le M, 1 \le n \le N; \tag{17}$$

$$\delta_{j,m,n}, \gamma_{j,m,n}, \zeta_{j,m,n} \in \{0, 1\}, \quad \forall j \in \{1, 2, \cdots, J_n\} \text{ and } m \in \{1, 2, \cdots, M\}; \tag{18}$$

$$\delta_{j,m,n} = 0, \gamma_{j,m,n} = 0, \zeta_{j,m,n} = 0, \quad \forall j \notin \{1, 2, \cdots, J_n\} \text{ or } m \notin \{1, 2, \cdots, M\}. \tag{19}$$

The objective is to minimize the discrepancy between binary indicators $P_{j,m,n}$ and binary decision variables $\delta_{j,m,n}$. Constraints (1) and (2) indicate that the crash took place on one of the N candidate links. Constraints (3) through (7) describe the constraints on the originating cell of the crash. Constraints (8) through (12) describe the constraints on the terminating cell of the crash. Constraints (13) and (14) restrict the minimum requirements on the maximum temporal and spatial extent of the crash, respectively. Constraints (15) restrict the maximum amount of allowable correction in the vicinity of the reported occurrence time and location. Constraints (16) through (17) restrict the propagation of shockwaves. Note that Constraints (17) are identical to Constraints (6). Therefore, Constraints (6) are excluded from our final optimization model. Constraints (18) restrict the decision variables to be binary. Constraints (19) define the boundary conditions when subscripts j and m may exceed their respective ranges. The optimization model can be efficiently solved by the standard branch-and-bound algorithm.

We prove that the model has the following properties: (1) The solution to our optimization model produces the impact region for exactly one of the candidate links; (2) Let us assume that the impact region is produced for link n' and in the optimal solution there exists $1 \leq j' \leq J_n$ and $1 \leq m' \leq M$ such that $\gamma_{j',m',n'} = 1$. We then have that the impact region produced in the speed contour plot of link n' originates from cell $\langle j', m' \rangle$ and its shape conforms to the propagation of shockwaves; and (3) Let us assume that the impact region is produced for link n' and in the optimal solution there exists $1 \leq j'' \leq J_n$ and $1 \leq m'' \leq M$ such that $\zeta_{j'',m'',n'} = 1$. We then have that the impact region produced in the speed contour plot of link n' terminates at cell $\langle j'', m'' \rangle$.

3 Numerical Experiments

In this section, we validate the performance of our model using real crash data in Beijing. We obtain the crash report for crashes occurred on North 3^{rd} Ring Road in Beijing in the month of April 2016. This road is 10 kilometers in length and altogether there are 19 crashes in this month. The major issues with the crash report are: (1) The direction of travel is missing in all location descriptions . As a result, we are not able to know whether it occurred on the westbound link or the eastbound link; (2) The location description can be rather imprecise; and (3) The recorded reported time is actually the time when the call was received by the traffic police, which is certainly later than the actual occurrence time of the crash.

For each reported crash, we first obtain the coordinates, that is, the longitude and latitude, of the location that corresponds to its textual description by calling the geocoding function of Google Map [11]. We then draw an error circle with a radius of 100 meters to identify the candidate links. For each of the links, say link n, we construct its speed contour plot as follows: (1) We discretize time into 5-minute intervals and distance into 100-meter sections. This implies that the resolution of the speed contour plot is 100 meters $\times$ 5 minutes; (2) The analysis period is 2.5 hours long, which starts 0.5 hours prior to and ends 2 hours after the reported crash time; and (3) We obtain the mean and standard deviation of the crash-free speed ($\bar{s}_{j,m,n}$ and $\sigma_{j,m,n}$), the crash-induced speed matrix ($\hat{s}_{j,m,n}$), and $P_{j,m,n}$ according to the steps detailed in [13]. The minimum temporal and spatial extents of a crash are set to 30 minutes and 1 kilometer, respectively, which means $\Theta = 30/5 = 6$ and $\Phi = 1/0.1 = 10$. The maximum amount of allowable corrections to the reported occurrence time (m^*) and the reported crash location (j_n^*) are set to 30 minutes and 0.5 kilometers, respectively, which means $\Lambda^- = \Lambda^+ = 30/5 = 6$ and $\Delta_n^- = \Delta_n^+ = 0.5/0.1 = 5$.

The average performance across all 19 crashes shows that: (1) Our model can reduce the average bias in time from 7.8 minutes to 1.7 minutes, which is a 78.21% reduction; (2) Our model can reduce the average bias in location from 0.140 kilometers to 0.025 kilometers, which is a 82.14% reduction; and (3) The distance-based technique can get the correct direction of travel 10/19 = 52.63%, which is improved to 19/19=100% by our model.

4 Conclusions

In this research, we propose to simultaneously correct the time and location bias associated with a reported crash, which is new to the literature. For a given crash, we first follow the procedure detailed in [8] to identify the set of candidate links in the vicinity of the reported crash site. We then examine the spatiotemporal evolution of travel speed on these candidate links and select the one that is most congruent with the occurrence of a crash. We formulate the candidate selection process as an integer programming model and develop a set of novel constraints to estimate the spatiotemporal impact regions, which characterize the evolution of speed in the speed contour plots of the candidate links. We finally use the time and location when travel speed begins to drop to correct the time and location bias associated with the crash. We validate our model using real crash data in Beijing and find that our model can reduce the average bias in time from 7.8 minutes to 1.7 minutes, or a 78.21% reduction; and reduce the average bias in location from 0.140 kilometer to 0.025 kilometer, or a 82.14% reduction.

References

[1] Chao Wang, Mohammed A Quddus, and Stephen G Ison. Impact of traffic congestion on road accidents: A spatial analysis of the m25 motorway in england. *Accident Analysis & Prevention*, 41(4):798–808, 2009.

[2] Marianna Imprialou and Mohammed Quddus. Crash data quality for road safety research: Current state and future directions. *Accident Analysis & Prevention*, 2017.

[3] Paul A Zandbergen. A comparison of address point, parcel and street geocoding techniques. *Computers, Environment and Urban Systems*, 32(3):214–232, 2008.

[4] Kevin Austin. The identification of mistakes in road accident records: Part 1, locational variables. *Accident Analysis & Prevention*, 27(2):261–276, 1995.

[5] Ned Levine, Karl E Kim, and Lawrence H Nitz. Spatial analysis of honolulu motor vehicle crashes: I. spatial patterns. *Accident Analysis & Prevention*, 27(5):663–674, 1995.

[6] Becky PY Loo. Validating crash locations for quantitative spatial analysis: A gis-based approach. *Accident Analysis & Prevention*, 38(5):879–886, 2006.

[7] Lipika Deka and Mohammed Quddus. Network-level accident-mapping: distance based pattern matching using artificial neural network. *Accident Analysis & Prevention*, 65:105–113, 2014.

[8] Maria-Ioanna M Imprialou, Mohammed Quddus, and David E Pitfield. High accuracy crash mapping using fuzzy logic. *Transportation Research Part C*, 42:107–120, 2014.

[9] Maria-Ioanna M Imprialou, Mohammed Quddus, and David E Pitfield. Multilevel logistic regression modeling for crash mapping in metropolitan areas. *Transportation Research Record: Journal of the Transportation Research Board*, (2514):39–47, 2015.

[10] Andrew Tarko, Jose Thomaz, and Darion Grant. Probabilistic determination of crash locations in a road network with imperfect data. *Transportation Research Record: Journal of the Transportation Research Board*, (2102):76–84, 2009.

[11] Shaun Burns, Luis Miranda-Moreno, Joshua Stipancic, Nicolas Saunier, and Karim Ismail. Accessible and practical geocoding method for traffic collision record mapping: Quebec, canada, case study. *Transportation Research Record: Journal of the Transportation Research Board*, (2460):39–46, 2014.

[12] Younshik Chung. Quantification of nonrecurrent congestion delay caused by freeway accidents and analysis of causal factors. *Transportation Research Record: Journal of the Transportation Research Board*, (2229):8–18, 2011.

[13] Zhengli Wang, Xin Qi, and Hai Jiang. Estimating the spatiotemporal impact of traffic incidents: An integer programming approach consistent with the propogation of shockwaves. *Transporation Research Part B*, 111(12):356–369, 2018.

[14] Younshik Chung and Wilfred W Recker. A methodological approach for estimating temporal and spatial extent of delays caused by freeway accidents. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1454–1461, 2012.

Estimating Vehicle Fleet Composition for Last-Mile Delivery Service

Ekaterina Alekseeva
Colisweb

Luce Brotcorne
Inria-Lille Nord Europe

Youcef Magnouche
Inria-Lille Nord Europe

Etienne Soufflet
Colisweb

Frédéric Semet
Univ. Lille, CNRS, Centrale Lille, Inria UMR 9189 - CRIStAL
Lille, France

Email: frederic.semet@centralelille.fr

1 Problem description

In this presentation, we address a fleet composition problem (FCP) faced by last-mile delivery service companies (LMDSC), such as Colisweb, that are middlemen between e-commerce companies and carriers. LMDSCs organize transportation services for e-commerce companies and take advantage of higher volumes to mutualize more efficiently the transportation part. LMDCs, as Colisweb, do not manage their fleet but have contracts with local carriers. Carriers propose different types of vehicles (bicycles, motorcycles, cars, and vans) with different transportation costs per km. One day in advance, LMDSCs have to decide how many vehicles of each type will be needed to cover the transportation demand.

One of the main characteristics of the problem addressed is that the demand is not known a priori, due to the inherent uncertainty of this type of activity. However, the demand is approximated as follows. The distribution area is divided into a limited number of delivery zones and the time horizon into time slots. Demand is characterized by a forecast number of packages to be transported from pickup zones to delivery zones within a delivery time slot.

Additional constraints such as vehicle capacities, the maximum working time have to be taken into account. The FCP, we tackle, consists in determining the minimum cost vehicle fleet to cover the demand while satisfying such side-constraints. The total cost is computed as the sum of han-

dling costs and of traveling costs. Several variants of the FCP have been studied in the literature. In [1], authors present an interesting survey on this family of problems and distinguish between the strategic and the tactical/operational FCPs. In tactical/operational FCPs, the decisions are related to an assignment of vehicles to routes or to transportation operations. A well-known tactical/operational FCP is the fleet size and mix vehicle routing problem (FSMVRP). The variant of the FCP, we deal with, also belongs to this subfamily, but differs from the FSMVRP since the locations of the pickup and delivery points remain unknown.

We now provide a precise description of the FCP tackled and some notation. An LMDSC subcontracts the transportation of packages to a set of carriers. Each carrier owns different types of vehicles with characteristics such as capacities, speed and traveling cost. Let V be the set of types of vehicles and R be the set of resources (volume capacity, carrying capacity, maximum number of packages). Let b_v^i be the capacity of vehicle type $v \in V$ associated with resource $i \in R$. A courier, associated with each vehicle, works at most Ω minutes per day.

The delivery area is divided into zones which do not intersect. Zones correspond typically to subsets of adjacent postal codes. The speed of each vehicle varies according to the zones. Let ϑ_z^v be the traveling speed of a vehicle of type v in zone z. For two different zones z, z', $f_{zz'}$ is the distance between the zone z and z'.

The packages have to be delivered over a horizon time T, which is divided into *time slots*. All time slots have the same duration δ_t, and do not intersect. For a time slot t, s_t and e_t are the starting and ending times. For two time slots t, t', we denote by $t < t'$ if $e_t \leq s_{t'}$. In the following, we assume that each vehicle visits at most one zone during one time slot. c_{vzt}^{km} is the cost per km associated with a vehicle of type v when it is in zone z during time slot t.

We suppose that the transportation demand is unknown and is approximated as follows. Let D be the set of transportation requests. For each zone z, let D_z (resp. $\overline{D}_z$) $\subseteq D$ be the set of requests with the pickup (resp. delivery) locations in zone z. For each request d, let r_d^i be the consumption of resource $i \in R$, and $[s^d, e^d]$ be the delivery time window. There is no time-window for the pickup, which has only to be performed before the delivery. From the delivery time-window, we determine T_d (resp. $\overline{T}_d$) the set of possible time slots to perform the pickup (resp. the delivery) of request d. Let z_d (resp. $\overline{z}_d$) be the pickup (resp. delivery) zone of request d, and t_d^+, t_d^- be the pickup and delivery durations. Last, we denote by Let $c_{vtd}^{pickup}, c_{vtd}^{delivery}$ the costs for the pickup and delivery operations of request d using a vehicle of type v during time slot t.

The FCP, we deal with, consists in assigning each request to a vehicle and to pickup and delivery time slots such that: i) the pickup is done before the delivery; ii) the capacities of each vehicle are respected; iii) each vehicle can visit at most one zone per time slot; iv) the total time of operations in a time slot does not exceed its duration; v) the total working time per vehicle does not exceed Ω, and the total cost is minimized.

2 Extended formulation and solution methods

First, we developed a compact integer linear formulation for the FCP described above. By lack of space, this model is not provided here. We then derived from it, the extended formulation, we present here, by applying a Dantzig-Wolfe decomposition. The master problem can be expressed as follows. Given a set of requests $\mathcal{D} \subseteq D$, let $\mathcal{A}_\mathcal{D}$ (resp. $\overline{\mathcal{A}}_\mathcal{D}$) be an assignment of each request in $\mathcal{D}$ to a time slot in T for the pickup (resp. delivery) operation. Let $c_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}}$ be the total cost when the requests in $\mathcal{D}$ are allocated to a vehicle of type v and executed according to the time-slot assignments $\mathcal{A}_\mathcal{D}$, $\overline{\mathcal{A}}_\mathcal{D}$, and $a_d^{\mathcal{A}_\mathcal{D}t}$ equals to 1 if d belongs to $\mathcal{D}$ and the delivery is performed during the time slot t according to the assignment $\overline{\mathcal{A}}_\mathcal{D}$, 0 otherwise. We define binary variables $\lambda_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}}$ be equal to 1 if a vehicle of type v covers the requests in $\mathcal{D}$ with assignments $\mathcal{A}_\mathcal{D}$ and $\overline{\mathcal{A}}_\mathcal{D}$, 0 otherwise. The FCP is formulated as:

$$\min \sum_{\mathcal{D}\subseteq D}\sum_{\mathcal{A}_\mathcal{D}}\sum_{\overline{\mathcal{A}}_\mathcal{D}} c_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}}\lambda_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}}$$

$$\sum_{\mathcal{D}\subseteq D|d\in\mathcal{D}}\sum_{\mathcal{A}_\mathcal{D}}\sum_{\overline{\mathcal{A}}_\mathcal{D}}\sum_{t\in T_d} a_d^{\overline{\mathcal{A}}_\mathcal{D}t}\lambda_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}} = 1, \quad \forall d \in D, \qquad (\beta_d) \tag{1}$$

$$\lambda_v^{\mathcal{A}_\mathcal{D}\overline{\mathcal{A}}_\mathcal{D}} \in \{0,1\}.$$

In this model, each request has to be assigned to a vehicle while minimizing the total cost. β_d represents the dual variable associated with constraint (1). The pricing problem per type of vehicle is modeled as a binary linear program using the following variables. Let $x1_d^t$ (resp. $x2_d^t$) be equal to 1 if d is picked-up (resp. delivered) during time slot t, and 0 otherwise, and let $y_{zz'k}$ be equal to 1 if the vehicle is in zone z during time slot t and in z' during time slot $t+1$, 0 otherwise. u_z^k is equal to 1 if the vehicle is in zone z during time slot t, 0 otherwise. L and U represent the starting and ending times for the operations performed with the vehicle. The pricing problem associated with a vehicle of type v is formulated as:

$$\min \quad \sum_{d\in D}\Big(\sum_{t\in T_d}\Big(\frac{c_{vz_dt}^{km} l_{z_d}}{6} + c_{vtd}^{pickup}\Big)x1_d^t + \sum_{t\in\overline{T}_d}\Big(\frac{c_{v\overline{z}_dt}^{km} l_{\overline{z}_d}}{6} + c_{vtd}^{delivery} - \beta_d\Big)x2_d^t + \sum_{t\in T}\sum_{z\in Z}\Big(c_{vzt}^{km}(2l_z)y_{zz}^t + c_{vzt}^{km} f_{zz'}\sum_{z'\in Z} y_{zz'}^t\Big)\Big)$$

$$\sum_{t\in\overline{T}_d} x2_d^t \leq 1, \qquad \forall d \in D, \tag{2}$$

$$x2_d^t \leq \sum_{t'\in T_d|t'\leq t} x1_d^{t'}, \quad \forall d \in D, t \in \overline{T}_d, \tag{3}$$

$$\sum_{d\in D}\Big(\sum_{t'\in T_d|t'\leq t} r_d^i x1_d^{t'} - \sum_{t'\in\overline{T}_d|t'\leq t} r_d^i x2_d^{t'}\Big) \leq b_v^i, \qquad \forall i \in R, t \in T, \tag{4}$$

$$\sum_{d\in D_z|t\in T_d}\Big(t_d^+ + \frac{l_z}{6\vartheta}\Big)x1_d^t + \sum_{d\in\overline{D}_z|t\in\overline{T}_d}\Big(t_d^- + \frac{l_z}{6\vartheta}\Big)x2_d^t + \frac{2l_z}{\vartheta}(1-y_{zz}^t) + \sum_{z'\in Z\setminus\{z\}}\Big(\frac{f_{zz'}}{2\vartheta}y_{zz'}^t + \frac{f_{z'z}}{2\vartheta}y_{z'z}^{t-1}\Big) \leq \delta_t, \quad \forall t \in T, z \in Z, \tag{5}$$

$$\sum_{z\in Z} u_z^t \leq 1, \qquad \forall t \in T, \tag{6}$$

$$u_z^t - u_z^{t+1} \leq y_{zz}^t, \qquad \forall t \in T, z \in Z \tag{7}$$

$$u_z^t + u_{z'}^{t'} - \sum_{z''\in Z}\sum_{t<t''<t'} u_{z''}^{t''} \leq y_{zz'}^t + 1, \quad \forall t, t' \in T, t < t', z \neq z' \in Z, \tag{8}$$

$$\sum_{d\in D_z|t\in T_d} x1_d^t + \sum_{d\in\overline{D}_z|t\in\overline{T}_d} x2_d^t \leq M u_z^t, \quad \forall t \in T, z \in Z, \tag{9}$$

$$U - L \leq \Omega, \tag{10}$$

$$U \geq e_t \sum_{z \in Z} u_z^t, \qquad \forall t \in T, \tag{11}$$

$$L \leq s_t \sum_{z \in Z} u_z^t + M(1 - \sum_{z \in Z} u_z^t), \qquad \forall t \in T, \tag{12}$$

$$x1_d^t, x2_d^t, y_{zz'}^t, u_z^t \in \{0,1\} \qquad U, L \in R^+. \tag{13}$$

where M is a large number. Note that the index v was omitted to lighten the mathematical expressions. Constraints (2) and (3) guarantee that each request is assigned at most once, and that the delivery can be performed if and only if the pickup took place before. Inequalities (4) represent the capacity constraints. Constraints (5) ensure that total time spent for pickup and delivery operations and traveling not exceed the duration of a time slot. Since we do not know the pickup and delivery locations within a zone, we approximate the traveling distance using the formula proposed by [2] and extended by [3] for city logistics. We assume that each zone z is a square with a side length equal to l_z. Constraints (6) guarantee that at most one zone is visited per time slot. Inequalities (7), (8) and (9) link variables y, x and u. Constraints (10), (11) and (12) ensure that the courier works at most Ω minutes. Last, the objective function corresponds to the reduced cost where the traveling cost is computed as in constraint (5).

We developed a branch-and-price algorithm as well as a diving heuristic to solve this model. The branch-and-price method includes several features such as: 1) preprocessing techniques to set some variables in the subproblem; 2) heuristics to solve the subproblem; 3) specific rules to generate several columns for a given type of vehicle at each iteration; 4) an adapted branching scheme. Last, we implemented a diving heuristic based on the branch-and-price algorithm. At each iteration, we set one variable of the restricted master problem to 1 and solve again the restricted master problem adding some columns. We reiterate until we obtain an integer solution.

To assess the efficiency of the proposed approaches, we solved real-life instances provided by our industrial partner. Here, we report a summary of average computational results when $|D|$ varies from 99 to 179, the number of zones is twenty, $\delta_t = 120$ minutes, and $|V|$ =1, 2, 3. When the CPU is limited to 1 hour, none instance can be solved to optimality with the compact formulation or using the branch-and-price algorithm. Depending on the number of vehicle types, the gaps between the best-known solutions and the best lower bounds vary between 8.41% and 9.55% for the compact formulation and between 2.87% and 2.91% for the branch-and-price algorithm. The diving heuristic performs quite well since the gaps are between 1.87% and 2.25% while the CPU times are between 342 and 603 seconds.

References

[1] A. Hoff, H. Andersson, M. Christiansen, G. Hasle and A. Løkketangen, "Industrial aspects and literature survey: Fleet composition and routing", *Computers & Operations Research* 37, 2041–2061 (2010).

[2] C. Daganzo, "The distance traveled to visit N points with a maximum of C stops per vehicle: An analytic model and an application", *Transportation Science*, 18, 331–350 (1984).

[3] A. Franceschetti, A sustainable city logistics: fleet planning, routing and scheduling problems, Ph.D. thesis, Technische Universiteit Eindhoven (2015).

Machine Learning → Mathematical Programming for Air Crew Scheduling

François Soumis

GERAD

Polytechnique Montréal, Canada

Email : francois.soumis@gerad.ca

Yassine Yaakoubi

Department of Mathematics and Industrial Engineering

Polytechnique Montréal, Canada

Simon Lacoste-Julien

Department of Computer Science and Operations Research

University of Montréal

1 The problem

The crew pairing problem is generally modelled as a set partitioning problem, the flights have to be partitioned in pairings. A pairing is a crew path starting at a base covering many flights during few days of works and finishing at the same base. This problem becomes difficult to solve when the number of flights increases because the number of feasible pairings grows exponentially (number of variables). This paper introduces a new paradigm for solving this large combinatorial problem: "Machine Learning → Mathematical Programming". This paradigm use Machine Learning to learn from solutions of similar instances to produce predictions on some parts of the solution of the new instance. This information feeds the Mathematical Programming optimizer to finish the work taking account of the exact cost function and the complex constraints. This approach reduces significantly the solution time without losing on the quality of the solution.

2 State of the art

The most prevalent method since the 1990s to solve this large set covering problem is the column generation inserted in branch-&-bound, see [1], [2]. This method is described with others in a recent survey [3] concluding than column generation is the most frequently used approach. The column generation method iterates between pairing generation (sub problem (SP)) and pairing selection (master problem (MP)). For the SP many authors have used shortest-path in a network with resource constraints [4], [5]. The MP is a set partitioning problem [6] and the access to all legal pairings reduces deeply the

integrity gap and permits to obtain very good integer solutions with a partial exploration of the branching tree for problems with few thousand flights.

When the number of flights increases in a crew pairing problem, the time to solve it by column generation becomes large. The number of iterations of the column generation, the time per iteration for solving the master problem and the number of branching nodes increase. The Dynamic Constraints Aggregation method (DCA) developed by [7] speed-up the master problem by reducing the number of constraints and the degeneracy. The DCA method start with an aggregation, in clusters, of flights having a good probability to be done consecutively by the same crew, in an optimal solution. It corresponds to temporarily fix to one some flight connection variables. This permit to replace all the flight covering constrains of the flights in a cluster by a single constrain. DCA use reduces costs to identify flight connection variables that can be unfixed to improve the solution by breaking clusters. At the opposite, it aggregate clusters connected by connection variables equal to one. This dynamic management of the clusters aggregating the constraints permit to reach an optimal solution with a smaller and less degenerated master problem. This method produces better dual variables and reduces the number of column generation iterations. Furthermore, the LP solution is less fractional and it reduces the number of nodes to explore in the branch and bound.

Application on 10 000 flights weekly problems was presented at TRISTAN 2016 [8]. DCA was used to improve solutions produced by GENCOL a column generation solver embedded in a rolling horizon approach that was used by 20 airlines at this time. With windows of two days and one day overlaps, 6 problems of 3000 flights need to be solved and it took 40 hours. Starting with a partition defined by this initial solution DCA improves the solution of 1.21% in 5 hours.

3 The new solution method

To produce initial clusters, we use machine learning to determine some flight connection variables that will probably be equal to one. These clusters are not a feasible solution, but DCA will repair it with phase 1 and phase 2 of the simplex to reach a good feasible solution.

Using several months of historical crew pairing data covering tens of thousands of flights per month, we build a flight-connection prediction sub-problem in which the goal is to predict the next flight that an airline crew should follow in their schedule. To avoid error propagation through the entire sequence of flights, we only use information about the incoming flight of the crew to predict the next flight. Each flight is described by the city of origin and destination, the aircraft type and tail number, the flight duration, and the time of departure and arrival.

The multiclass classification problem is thus framed as follows: given the information about an incoming flight for a crew in a specific connecting city, choose the flight that the crew should follow among all of the possible departing flights from that city. We do not use the flight code since it is not stable information from month to month. Some flight codes appear or disappear each month. The schedule of a flight code can be changed, permuting the departure order of the flights. We find the next connecting flight based on the flight parameters instead of flight codes.

Due to the constraints-based nature of the learning problem, we can use *a priori* knowledge to define which flights are feasible [12]. For example, for an incoming flight, it is not possible to make a flight that starts ten minutes after the arrival, nor it is possible five days later. It is also rare that the type

of aircraft changes between flights, since each aircrew is formed to use one or two types of aircraft at most (see [9]). Therefore, for each incoming flight, we consider the feasible departing flights to be in the next 48 hours, according to inclusive constraints. We sort the flights based on the time of departure, and limit the maximal number of possible flights to 20, as this is sufficient in the airline industry. Thus, we predict the rank of the true flight among that set of flights.

In about 90% of the cases in our dataset, the aircrew arrives at an airport on an incoming flight and follows the aircraft, taking the next departing flight by the same aircraft. To consider more difficult next-flight prediction instances, we report not only the overall accuracy, but also the accuracy for only the instances in which the crew changes aircraft; we call this the "Different aircraft accuracy" (D_acc). We use convolutional neural networks where different hyperparameters (optimizer, learning rate, number of layers, etc.) [10] are optimized using Bayesian optimization, which comprises a few random searches followed by the standard Gaussian process optimization [11]. To explore the hyperparameters space, we utilize *k*-fold cross-validation, using separate months as different folds to simulate a realistic scenario in which we make a prediction on a new time period. We maintain one weekly problem (10,000 flights) for testing.

4 Computational experimentation

The experiments are executed on a 40-core machine with 384 GB of memory. Each model is executed in an asynchronously parallel setup of 2-4 GPUs. That is, it can evaluate multiple hyperparameter configurations in parallel, with each one on a single GPU. For a given hyperparameter configuration, the learning phase takes approximately ten minutes, and predicting the most probable next flights for a weekly problem only takes a few seconds. After a few iterations, we are able to obtain an accuracy of 99.35% (71.79% D_acc). Then, a random search raises the accuracy to 99.62%. Using a Gaussian process, we show that we continuously improve our process of searching for the best architecture to maximize the overall accuracy. In our case, we stop at iteration 500 with the best architecture providing an accuracy of 99.68% (82.53% D_acc), but it should be noted that there is no limitation to prevent our algorithmic procedure from further exploring.

Upon the finalization of the flights-connection prediction model, we use the best identified architecture to solve two other prediction problems: (i) predict if each of the scheduled flights is the beginning of a pairing or not; and (ii) predict whether each flight is performed after a layover or not. Using these predictors on a weekly problem, we construct some crew pairings and use them as initial clusters for the GENCOL-DCA solver. Unfortunately, if a flight in the pairing is poorly predicted, it can become impossible to construct a legal pairing finishing at the base. Therefore, we propose several heuristics to build pairings in such a manner that they always finish at the base.

To compare the results, we consider a benchmark obtained with the GENCOL solver by rolling horizons with two-days windows (GENCOL init.) and a solution obtained by using the GENCOL solution as an initial cluster for the constraint aggregation (GENCOL-DCA). Two of the three heuristics that we propose outperform the benchmark as well as the solution from GENCOL-DCA with GENCOL initialization, and we can conclude that for the pairings that finish away from the base, it is better to allow the solver to cover the flights than to propose smaller pairings. Therefore, instead of performing the optimization process for 40 hours to obtain GENCOL init. and then for another 5 hours to obtain

GENCOL-DCA, our proposed method gives better costs after a few seconds to predict the flight connections, and optimized results after five to six hours.

Approach	Savings (%)	Time of execution	Number of deadheads	Number of fract. variables at N0	Number of nodes	Number of GENCOL iter.
GENCOL init.	0	40:00	45			
GENCOL-DCA from GENCOL init.	1.21	05:00	27	1603	59	1196
GENCOL-DCA from heuristic 1 init.	1.45	06:30	22	1548	46	1191
GENCOL-DCA from heuristic 2 init.	1.38	05:00	24	1755	54	1395
GENCOL-DCA from heuristic 3 init.	0.97	06:30	28	1508	45	1196

Table 1: Final crew pairing costs after running GENCOL-DCA with different initialization clusters.

Furthermore, the number of deadheads is deeply reduced with DCA reoptimizing a full week simultaneously. It permits to remove some flights from a base and construct pairings from another base to satisfy base constraints. GENCOL-DCA from heuristic 1 and 2 does the best on this point and produce the best savings. Note than a saving of 1% on a crew cost of 2–3 billions per year is very significant. Observe also the very good performance of the optimizer using the initial clusters provided by Machine Learning for this problem of more than 10 000 constraints: there are only 1600-1700 fractional variables at node zero, 50 nodes in the branch and bound tree and 1200 to 1400 GENCOL iterations including re-optimizations in the branching tree. The performance does not suffer from the fact that the initial clusters are not a feasible solution.

We will present at the conference results on monthly problems solved by rolling horizon with one-week windows. The fast Machine Learning predictor will permit to construct in few seconds clusters for each window of a week, customized according to the flight schedule of this week. We expect improvements in particular when the monthly schedule is irregular from week to week. It is the case near every month: Christmas, Easter, Thanksgiving, National Holiday, Mother and Father days, big sport events (Superball, ….), … It was out of question to use five times 40 hours to produce customized clusters for each window with GENCOL init.

5 Conclusion

It is rather difficult to assign the crew workers to a range of tasks while taking into account all the variables and constraints associated with the process. We focus on the problem of predicting the next flight of a crew, framed as a multiclass classification problem trained from historical data, and adapt a neural network solution by reducing the number of classes to predict using domain-appropriate constraints, achieving a high accuracy (99.7% overall or 82.5% on harder instances).

This paper introduces a new paradigm for solving a large combinatorial problem: “Machine Learning → Mathematical Programming”. This paradigm, first, use Machine Learning to learn from solutions of similar instances to produce predictions on some parts of the solution of the new instance. This information feeds the Mathematical Programming optimizer to finish the work taking account of the exact cost function and the complex constraints. This approach reduces significantly the solution time without losing on the quality of the solution. This new paradigm can be applied on many types of problems solved recursively.

References

[1] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M.M. Solomon and F. Soumis, "Crew pairing at Air France", *European Journal of Operational Research* 97, 245-259 (1997).

[2] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh and P.H. Vance, "Branch-and-price: Column generation for solving huge integer programs", *Operations Research* 46, 316-329 (1998).

[3] M. Deveci and N.C. Demirel, "A survey of the literature on airline crew scheduling", *Engineering Applications of Artificial Intelligence* 74, 54-69 (2018).

[4] S. Lavoie, M. Minoux and E. Odier, "A new approach for crew pairing problems by column generation with an application to air transportation", *European Journal of Operational Research* 35(1), 45-58 (1988).

[5] J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis, "Time constrained routing and scheduling", in *Handbooks in Operations Research and Management Science*, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), 35-139, Elsevier, 1995.

[6] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. Solomon and F. Soumis, "Crew pairing at Air France", *European Journal of Operational Research* 97(2), 245-259 (1997).

[7] I. Elhallaoui, D. Villeneuve, F. Soumis and G. Desaulniers, "Dynamic aggregation of set partitioning constraints in column generation", *Operations Research* 53, 632-645 (2005).

[8] F. Soumis, M. Saddoune, F. Lessard, Proceeding, TRISTAN, Aruba (2016).

[9] A. Kasirzadeh, M. Saddoune and F. Soumis, "Airline crew scheduling: models, algorithms, and datasets", *EURO Journal on Transportation and Logistics* 6, 111-137 (2017).

[10] F. Dernoncourt and J. Y. Lee, "Optimizing neural network hyperparameters with Gaussian processes for dialog act classification", in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 406-413, 2016.

[11] C.E. Rasmussen and C.K.I. Williams, *Gaussian processes for machine learning,* MIT Press, 2008.

[12] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic and S. Lacoste-Julien, "Unsupervised learning from narrated instruction videos", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4575-4583, 2016.

The impact of road capacity on connectivity by eigenvector centrality analysis

Hiroe Ando

Graduate School of Engineering

Gifu University

Email: hiroe@gifu-u.ac.jp

Michael G.H. Bell

Institute of Transport and Logistics Studies

University of Sydney Business School

Fumitaka Kurauchi

Faculty of Engineering

Gifu University

1 Introduction

Various indicators and methodologies for measuring road network connectivity have been developed in past research (e.g., [1], [2]). However, almost all approaches require estimates of link traffic volumes and/or OD travel time obtained by a time-consuming traffic assignment work. Traffic assignmemt is difficult to apply to large road networks because of its heavy computational load, and OD travel time in real traffic cnditions is occasionally inaccurate.

This paper applies a capacity weighted eigenvector centrality method to identify the strongly and weakly connected parts of the network without referring to demand information or traffic assignment. This paper compares the results of unweighted and capacity weighted network to verify the advantages of the evaluation by capacity weighted eigenvector centrality.

In previous works [3], [4], the spectral partitioning method was applied to large road networks to identify the cut set with minimum capacity. Also, [4] applied various weight settings to see the difference depend on them. However, the spectral partitioning method is only applicable to undirected network. One attraction of the eigenvector centrality measure is for a directed network that can consider bidirectional movement.

2 Eigenvector centrality

The eigenvector centrality was defined in 1972 [5]. The eigenvector centrality is the value of the eigenvector corresponding to the largest eigen value of an adjacency matrix of the network. The concept

of this method is that node importance is increased by having connections to other nodes that are themselves important.

Let

$$\mathbf{Ax} = \lambda \mathbf{x} \tag{1}$$

where **x** is an eigenvector, λ an eigenvalue and **A** is a capacity-weighted adjacency matrix with element

$$a_{ij} = \begin{cases} \text{capacity of the link from node } i \text{ to node } j \\ 0 \text{ otherwise} \end{cases}$$

The Rayleigh quotient is

$$\lambda = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\sum_{i,j} x_i a_{ij} x_j}{\sum_i x_i^2} \tag{2}$$

We are interested in the largest eigenvalue (denoted by *)

$$\lambda^* = \max_{\mathbf{x}} \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\mathbf{x}^{*T} \mathbf{A} \mathbf{x}^*}{\mathbf{x}^{*T} \mathbf{x}^*} = \frac{\sum_{i,j} x_i^* a_{ij} x_j^*}{\sum_i x_i^{*2}} = \frac{\cdots + x_i^* a_{ij} x_j^* + \cdots}{\cdots + x_i^{*2} + x_j^{*2} + \cdots} \tag{3}$$

where $\mathbf{x}^*$ represents for the eigenvector corresponding to λ^*.

In this study, the connectivity of road network is evaluated by using the value of $\mathbf{x}^*$. Proofs are omited in exetnded abstract, if every node can be reached from any other node in the nework, then $\mathbf{x}^* > \mathbf{0}$ (or alternatively $\mathbf{x}^* < \mathbf{0}$). Besides, the value of the eigenvector corresponding to the largest eigen value does not require **A** to be symmetry.

3 Comparison of weight settings

The eigenvector centrality measure with small calculation load is tested to be applicable to several real road networks. To compare the results between unweighted and capacity weighted network, this study

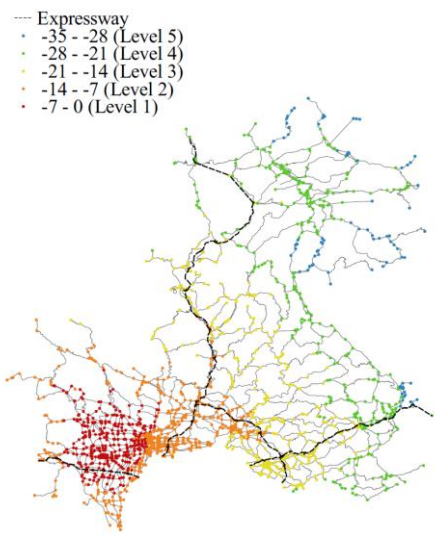


Fig.1 Eigenvector centrality (Unweighted)

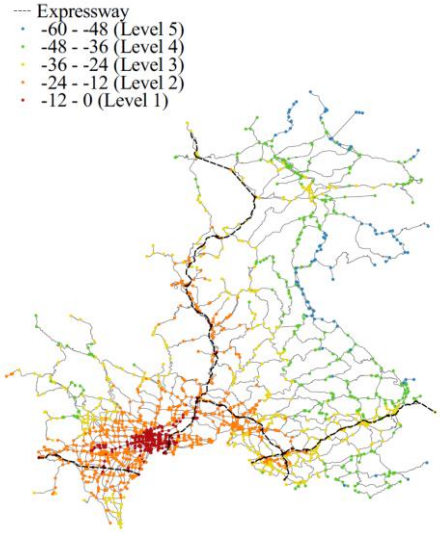


Fig.2 Eigenvector centrality (Capacity weighted)

uses Gifu Prefecture road network in Japan as an example. The Gifu Prefecture road network includes intercity expressways, national highways and the prefectural roads, contains 3,183 nodes and 9,482 links. Also, the elements of the adjacency matrix are directional capacities as mentioned earlier. In this network, the minimum and maximum link capacities are 1,000 and 80,000 vehicles per day, respectively.

Fig.1 and Fig.2 show the values of eigenvector centrality in unweighted and capacity weighted network, and the black dot line shows expressways. The values in the figure are shown on a log scale. Also, it is classified into 5 levels by dividing the range of eigenvalue centrality score after logarithm transformation into five equal intervals.

In the unweighted network which evaluates only by the topology of the road network, Level 1 nodes are located in the western part of Gifu Prefecture. From there, the connectivity is gradually becoming weaker towards the east. On the other hand, the result of the capacity weighted network shows that Level 1 nodes are located in a little eastern side, and the nodes with better connectivity spread along the expressways, that has larger link capacities. This difference is the impact of considering road capacity on eigenvector centrality.

The number of nodes in each level is similar in both networks. The level with the greatest number of nodes is Level 2, the lowest number of nodes is Level 5. However, Fig.3 shows the existence of nodes with different levels in both networks. There are several nodes with a high level in unweighted network and low level in capacity weighted network, and vice versa. Where are the nodes with the large level difference located? Fig.4 shows the nodes with large level difference between two networks. At first, nodes with lower levels in unweighted network than in capacity weighted network are obviously located along the expressway. This means that the capacity weighted eigenvector centrality clearly shows the influence of the large capacity road like expressways. Moreover, that nodes which have large level difference are not only lying on the expressways but also spread to several adjacent nodes. From this result, the capacity weighted network can take into account the ripple effect of roads with larger capacities (expressways). On the other hand, nodes with higher level in unweighted network than level in capacity weighted network, especially nodes with a difference of 2 levels or more are located mainly in the mountaineous area. Gifu Prefecture, however, also have large mountain areas in the northern part. Therfore, the reasons why these areas are picked up is not only because this. It can be said that although

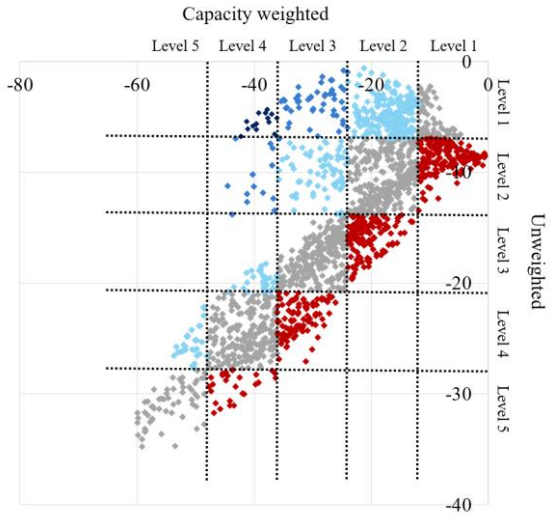


Fig.3 Eigenvector centrality on both networks

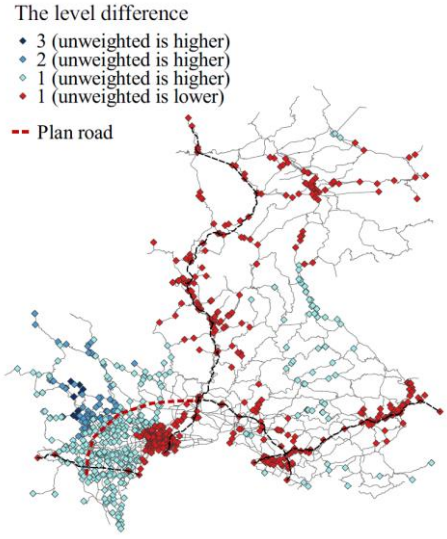


Fig.4 The difference of level between both networks

these areas have high conectivity from the topological point of view, but they are insufficient from the viewpoint of capacity.

Based on these results shown in this chapter, the level difference between unweighted and capacity weighted network mainly occurs along the expressway in the case study of Gifu Prefecture. Since these results make sense for the real road situation, the capacity weighted eigenvector centrality can evaluate effect of connectivity considering traffic capacity.

4 Conclustions

This paper applied the eigenvector centrality method to Gifu Prefecture road network to identify the strongly and weakly connected parts. Two settings of the weight are tested, unweighted network and capacity weighted network. From the comparison result of the two networks, the capacity weighted eigenvector centrality is found to be a useful measure for evaluating the connectivity considering traffic capacity.

For future tasks, it would be interesting to study the impact of link or node failures on road network as this may help to identify the large impact part. These results will be more persuasive by comparing to existing road network evaluation methods. Moreover, new expressways constructions are planned with red line shown in Fig.4. The areas where are insufficient from the viewpoint of capacity is near the planned expressways. Therfore, the connectivity of these areas may be improve, future works will focus on changes associated with new expressways constructions.

References

[1] H. Wakabayashi and Y. Iida, “Upper and lower bounds of terminal reliability of road networks: an efficient method with Boolean algebra”, *Journal of Natural Disaster Science 14*, 29-44, 1992.

[2] F. Kurauchi, N. Uno, A. Sumalee and Y. Seto, “Network evaluation based on connectivity vulnerability", *Transportation and Traffic Theory: Golden Jubilee*, 637-649, 2009.

[3] M.G.H. Bell, F. Kurauchi, S. Perera and W. Wong, “Investigating transport network vulnerability by capacity weighted spectral analysis”, *Transportation Research Part B, 99*, 251-266, 2017.

[4] H. Ando, F. Kurauchi, S. Myoko, M.G.H. Bell and S. Perera, “Connectivity evaluation of the road network using spectral partitioning method”, 7th International Conference on Transport Network Reliability, 2018.

[5] P. Bonacich, “Factoring and weighting approaches to status scores and clique identification”, *Journal of Mathematical Sociology, 2,* 113-20, 1972.

Multi-Period Workload Balance in Last-Mile Urban Delivery

Yang Wang
Department of Industrial Engineering
Tsinghua University

Lei Zhao
Department of Industrial Engineering
Tsinghua University, Beijing, China
Email: lzhao@tsinghua.edu.cn

Martin Savelsbergh
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

In the past two decades, the world has witnessed fast pace of urbanization and rapid growth of e-commerce, which jointly post significant challenges in urban delivery. In China, more than 3 million express delivery personnel or couriers delivered over 40 billion packages in 2017. 61.7% of the couriers worked 8-12 hours per day and 24.7% worked over 12 hours per day, with their daily delivery ranging from 10 to 150 packages (Beijing Jiaotong University et al., 2016).

A courier's income consists of low (or no) basic wage plus piece work, which is highly (or solely) dependent on the number of packages delivered, which we refer to as *incentive workload*. Intuitively, couriers want to deliver as many packages as possible. On the other hand, the effort (e.g., travel distance, delivery time, etc.) to deliver the packages, which we refer to as *effort workload*, may vary significantly, depending on the proximity among the delivery addresses, restrictions in accesses to certain delivery addresses, etc. Naturally, for the same amount of delivery packages, a courier may want to spend as little effort as possible. Recognizing and balancing these two types of workload is critical in maintaining the morale of couriers to ensure high-quality and sustainable last-mile delivery services to customers. In practice, while courier dispatching (i.e., delivery package assignment) decisions are made daily, couriers may pay more attention to their overall workload balance over a period of time (e.g., a month), rather than the (im)balance on a daily base.

In the literature of (single-period) vehicle routing problems considering workload balance, many researchers choose the multi-objective approach (see Matl et al., 2017 for an excellent review). In Matl et al. (2017), the authors distinguish between equity metrics, equity functions, and equity objectives. With equity

metrics, workload measured by route length/duration/cost correspond to effort workload and workload measured by route load corresponds to incentive workload. A few researchers study vehicle routing problems with load balance (e.g., Bowerman et al., 1995; Kritikos and Ioannou, 2010), while most focus on vehicle routing problems with route length balance. The objectives include minimizing the total route length/duration/-cost and minimizing load or route length imbalance. Route length (im)balance criteria include maximum length/duration/cost among routes (Corberán et al., 2002), range of lengths/durations/costs among the routes (Jozefowiez et al., 2009), and other criteria (Halvorsen-Wearea and Savelsbergh, 2016). Baŋos et al. (2013a,b) study vehicle routing problems with either load or route balance.

There have been very few study on multi-period workload balance. Mourgaya and Vanderbeck (2007) study a tactical "regionalization" problem to divide customers into clusters, each served by one vehicle. The objective is to minimize the total distance between customer points belonging to the same cluster while enforcing a restriction on the maximum demand assigned to a vehicle (within a cluster). Gulczynski et al. (2011) study a periodic vehicle routing problem (PVRP) that minimizes the weighted sum of the total distance and the range of load among all the routes of all the vehicles within a planning horizon. Liu et al. (2013) study a periodic vehicle routing problem with time windows (PVRPTW) that minimizes the maximum route cost over the entire horizon. These three papers are all under the setting of deterministic demand.

In this paper, we study the multi-period dynamic courier dispatching problem in last mile urban delivery under stochastic demand over a planning horizon (e.g., a month) and model the problem as a Markov decision process. We use the number of delivery packages as the measure of incentive workload and delivery time as the measure of effort workload. We study two imbalance criteria, namely, maximum workload and range of the workload ($MaxI$ and $RangeI$ for incentive workload and $MaxF$ and $RangeF$ for effort workload). An optimal courier dispatching policy minimizes the expected total cost composed of operational cost (measured by the total delivery time) and penalty cost on the two types of workload imbalance over the planning horizon.

We employ the policy function approximation (PFA) approach of approximate dynamic programming (ADP). We study three sets of courier dispatching policies. Under policy π^{Max}, we rebalance the cumulative workload in each period using maximum workload as imbalance measure. Similarly, we reblance the cumulative workload in each period using range as the imbalance measure under policy π^{Range}. We also study a policy that assigns packages to couriers according to a pre-determined delivery territory based on average workload, denoted as π^{Fixed}.

In our numerical experiments, we generate test instances based on the real delivery network of an e-commerce company. Some customers have low demand distribution while others have high demand distribution because of their geographical and demographical characteristics. We also capture the weekly demand pattern with higher demand on weekends than on weekdays. We focus on the workload balance among

three couriers over a 30-days planing horizon. Besides the three policies we described above (i.e., π^{Max}, π^{Range}, and π^{Fixed}), we also benchmark with a policy π^{CVRP} that solve a capaciated vehicle routing problem (CVRP) without imbalance penalty and tries to rebalance the (cumulative) workload when assigning the resulting routes to the couriers.

Tab. 1 shows the preliminary results on policy comparison. We highlight the best results in each evaluation metric in bold. As expected, π^{CVRP} results in the minimum total and average delivery time (or effort). Among all policies, the differences in the total (or average) delivery effort are within about 0.22%, indicating that the manager can improve workload balance without increasing the overall (or average) workload of the couriers. Note that the total (or average) incentive (number of packages) are the same for all policies. Overall, π^{Range} results in the best results in most workload balance metrics (except maximum effort). We observe that π^{Fixed} results in the worst workload balance, which is intuitive, because a slight difference in the average demand between delivery territories can be cumulatively enlarged over the planning horizon. We also observe that the rebalancing in π^{CVRP} does achieve improved workload balance, compared to π^{Fixed}. (This research is working in progress.)

Table 1: Policy comparison

Evaluation metric	π^{Max}	π^{Range}	π^{Fixed}	π^{CVRP}
Total delivery effort (min)	60,600.4	60,779.7	60,651.0	**60,051.2**
Average effort (min)	20,200.1	20,259.9	20,217.0	**20,017.1**
Range of effort (min)	254.3	**115.1**	1,981.6	245.2
Maximum effort (min)	20,318.1	20,319.6	21,347.1	**20,143.9**
Minimum effort (min)	20,063.8	**20,204.5**	19,365.5	19,898.7
Average incentive	5404.8	5404.8	5404.8	5404.8
Range of incentive	71.9	**31.2**	341.9	64.8
Maximum incentive	5,437.2	**5,422.1**	5,616.0	5,436.1
Minimum incentive	5,365.3	**5,390.9**	5,274.1	5,371.3

Keywords: Last-mile urban delivery; workload balance; multi-period; policy function approximation; approximate dynamic programming

References

Baŋos, R., Ortega, J., Gil, C., Fernąndez, A., De Toro, F., 2013a. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. Computers & Industrial Engineering 65 (2), 286–296.

Baŋos, R., Ortega, J., Gil, C., Fernąndez, A., De Toro, F., 2013b. A simulated annealing-based paralle multi-objective approach to vehicle routing problems with time windows. Expert Systems with Applications 40 (5), 1696–1707.

Beijing Jiaotong University, Ali Research, Cainiao Network, 2016. Research report for national e-commerce logistics employees. Technical report.

Bowerman, R., Hall, B., Calamai, P., 1995. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. Transportation Research Part A: Policy and Practice 29 (2), 107–123.

Corberán, A., Fernández, E., Laguna, M., Marti, R., 2002. Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of the operational research society 53 (4), 427–435.

Gulczynski, D., Golden, B., Wasil, E., 2011. The period vehicle routing problem: New heuristics and real-world variants. Transportation Research Part E: Logistics and Transportation Review 47 (5), 648–668.

Halvorsen-Wearea, E. E., Savelsbergh, M. W. P., 2016. The bi-objective mixed capacitated general routing problem with different route balance criteria. European Journal of Operational Research 251 (2), 451–465.

Jozefowiez, N., Semet, F., G., T. E., 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. European Journal of Operational Research 195 (3), 761–769.

Kritikos, M. N., Ioannou, G., 2010. The balanced cargo vehicle routing problem with time windows. International Journal of Production Economics 123 (1), 42–51.

Liu, R., Xie, X., Garaix, T., 2013. Weekly home health care logistics. In: Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on. IEEE, pp. 282–287.

Matl, P., Hartl, R. F., Vidal, T., 2017. Workload equity in vehicle routing problems: A survey and analysis. Transportation Science 52 (2), 239–260.

Mourgaya, M., Vanderbeck, F., 2007. Column generation based heuristic for tactical planning in multi-period vehicle routing. European Journal of Operational Research 183 (3), 1028–1041.

A MIP formulation for the flexible rostering of ground personnel at an international airport

Juan Pablo Cavada
Department of Civil Engineering,
Universidad de Chile, Santiago, Chile
Email: jucavada@ing.uchile.cl

Cristián E. Cortés
Department of Civil Engineering,
Universidad de Chile, Santiago, Chile

Pablo A. Rey
Department of Industry
Universidad Tecnolgica Metropolitana, Santiago, Chile

1 Introduction

In this paper we present a mixed integer programming approach to aid the rostering process of the ground staff (hereinafter called employees) in Santiago de Chiles international airport. This formulation was later wrapped in a decision support system to be used by the most important ground handling company in the airport. Rostering and scheduling problems have generated much interest as shown in the surveys done by [1] and [2].

This study focuses on finding an optimal assignment of shifts for the operators of the ground handling services of one of the airport main providers. These services include among others: baggage handling, tugs, tractors and loaders operation, aircraft cleaning and refueling. In this problem we need to build a monthly schedule for all 850 employees of the company, distributed in 22 different positions.

Besides complying with all legal regulations, the roster must satisfy operational and union constraints. For example, the roster is required to schedule a minimum number of employees in each position. Another constraint states that some skills are needed in each shift. Also, some employees may have special schedule requests: such as planned vacations, trainings, or may be part time students. Furthermore, there are several union agreements that can either ban or enforce certain shifts combinations.

Each day is divided in three shifts: Opening (O), from 4 a.m. to 12 noon; Afternoon (A), from 12 noon to 8 p.m.; and Night (N), from 8 p.m. to 4 a.m. A feature of this case is that personnel required among shifts differ significantly (both in total number and needed skills). Therefore, using regular shift sequences (like in [3]) may lead to serious over/understaffing during critical days. It is also possible that, in some positions the

number of available employees is not enough to fully cover all the shifts. To address these problems, the developed model needs to generate flexible rosters while trying to balance the understaffed shifts, so that no one is much worst that the others. Finally, there are some "quality of life constraints that go beyond the required by law: first, the rostering must produce an equitable workload to all employees in a position. Specifically, every month each employee should have approximately the same number of opening, afternoon and night shifts. Second, all employees must have at least two consecutive free days twice each month[1].

2 Rostering Model

In this section we present the main features of the MIP based approach. Due to the size of the problem we split its solution into two models that are solved in sequence: a shift assignment model and an hour assignment model. The first model assigns a shift to each employee for each day of the month. The second model assigns to each employee the exact hour of entry to the shift. The latter is a much simpler model and would not be discussed in this abstract. Also, we will not discuss all the constraints of the model, as many of them are common in MIP rostering and scheduling formulations.

Let d, e, t and s be indices representing the days in the month, employees, working shifts to be assigned (O, A, N and F for an off day), and skills, respectively. The objective function minimizes the weighted sum of five terms: the first term is the relative understaffing of the worst shift in the month, r; the second term is the summation of the understaffing in each day and shift, l_{dt}; the third term is the sum of all assigned employees, y_{edt}; the fourth is the sum of all deviations with respect to the expected number of shifts for an employee in his position, ql_{et}, qu_{et}[2]. The last term sums the difference between the required number of employees with a certain skill and the number of employees that are actually assigned, ls_{sdt}.

$$\min \alpha r + \beta \sum_{d} \sum_{t} + \gamma \sum_{e} \sum_{d} \sum_{t} y^{e}_{dt} + \delta \sum_{e} \sum_{t} (ql^{e}_{t} + qu^{e}_{t}) + \epsilon \sum_{s} \sum_{d} \sum_{t} ls_{sdt} \quad (1)$$

Most of the constraints regarding the compatibility between shifts that can be assigned to an employee can be modeled as either a positive or negative pair of shift sequences. We say that two shift sequences have a positive relation if every time the first sequence of the pair appears, it must be followed by the second sequence in the pair. For example, whenever an employee were assigned four night shifts in a row he would have the next

[1] Chilean law requires only a free weekend

[2] For instance, if the expected number of night shifts for his position is 6 and he has 4 assigned then $ql_{et} = 2$.

three days off, so the pair of sequences (NNNN, FFF) has a positive relation. On the other hand, a pair of sequences has a negative relation if the first sequence cannot be followed by the second one. For instance, an employee that is assigned to an afternoon shift cannot work at an opening shift the following day because he would not have enough rest time. This means that the pair (A, O) has a negative relation.

Let (S_1, S_2) be a pair of shift sequences and $S_k[i]$ be the shift in the i-th position of the sequence k. For a given employee e and day d, we define the positive pairs constraints in (2) and the negative pair constraints in (3):

$$\sum_{i=0}^{|S_1|-1} y^e_{d+i,S_1[i]} \leq |S_1| - 1 + y^e_{d+|S_1|+j,S_2[j]} \qquad \forall j \in \{0, \ldots, |S_2| - 1\} \qquad (2)$$

$$\sum_{i=0}^{|S_1|-1} y^e_{d+i,S_1[i]} + \sum_{j=0}^{|S_2|-1} y^e_{d+|S_1|+j,S_2[j]} \leq |S_1| + |S_2| - 1 \qquad (3)$$

Using these types of constraints, the planner can easily include additional conditions in the model, without changing the general formulation. Even more, these pairs can be general to all employees or specific to a group of them. This idea was inspired be the work presented in [4]

3 Implementation and results

The model was implemented into a decision support system to be used by the company. This implementation was performed as a standalone program using Java and the Gurobi Solver. They were several challenges that had to be addressed in the deployment. First the required data needed was scattered though different departments of the company (human resources, operations, maintenance, etc.), so a new data transfer protocol had to be developed. Secondly, often the model would not be able to find feasible solutions, as there are many hard constraints. To solve this, a feasibility check model was developed, that identified and reported all infeasibilities, thus allowing the planners to decide how to proceed with them.

The rosters generated by the model greatly improve the manual solution build by the company planner; the following results are a comparison between our solution and a manual roster used by the company. This cases consists in 850 employees distributed in 22 positions.

First, we can ensure that no labor or union agreement is violated, in contrast with the manual roster that has 3% of employees with an illegal schedule. The overall understaffing in critical positions was reduced form an average of 6% to a 2%, while also reducing the number of employees that performed an extra shift from 62% to 54%. The total average

number of days off per employee was reduced from 9.8 to 9.6 (this means approximately 170 less free days in total).

However, this model also improved the quality of life of the employees, as in the proposed solution all of them have two consecutive days off at least twice a month. Using the model more special request from employees can be satisfied, increasing from 84% in the manual roster to 99%. Finally, the number of employees with balanced schedules was increased from 45% to 71% [3].

A second, very important result, is that the time required to build the whole roster was reduced from 2 weeks to less than 3 days, leaving the planner more time to finely tune the solution and review the more complex situations.

Acknowledgements

This study was funded by the Complex Engineering Systems Institute (CONICYT - PIA - FB0816) and Vicerrectoría de Investigación y Desarrollo (VID) de la Universidad de Chile, project code: ENL24/18.

References

[1] A. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens, and D. Sier, "An annotated bibliography of personnel scheduling and rostering", *Annals of Operations Research* 127(14), 21144, 2014 .

[2] J. Van den Bergh, J. Belien, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck. "Personnel scheduling: A literature review". *European Journal of Operational Research*, 226(3), 367385, 2013.

[3] T. Fahle and W. Vermohlen, "Fair Cyclic Roster Planning A Case Study for a Large European Airport". *Operations Research Proceedings*, 129-135, 2016.

[4] M. C. Ct, B. Gendron, and L. M. Rousseau. "Grammar-based column generation for personalized multi-activity shift scheduling". *INFORMS Journal on computing*, 25(3), 461-474, 2013.

[3] A schedule is considered balanced if all employees has the same number shifts of each type

A Priori Routing for Strategic Time Slot Management in Online Grocery Retailing

Thomas R. Visser
Erasmus University Rotterdam, Rotterdam, The Netherlands

Martin Savelsbergh
Georgia Institute of Technology, Atlanta, USA
Email: thomasrvisser@gmail.com, martin.savelsbergh@isye.gatech.edu

1 Introduction

We consider an emerging strategy for offering and managing time slots for attended home grocery delivery: a set of routes is generated *a priori* and customers are assigned a time slot based on their home location and these routes. In such an environment, customers are offered only a few time slot choices per week, but it greatly simplifies operations for the retailer. We develop a 2-stage stochastic programming approach for designing a set of a priori routes and time slot assignments to be used in such an environment.

More specifically, we consider a retailer that offers its online customers a small number of time slots during which a delivery can take place. The retailer has a fleet of identical vehicles to make deliveries. Each vehicle starts and ends its delivery route at the retailer's fulfillment center. For each of its customers, the retailer knows the delivery location, the order size, the service time, the revenue, and the order placement probability. Observe that the only stochastic feature in this setting is whether or not a customer places an order. In practice, a customer's order size and revenue are likely to be stochastic as well. Customers can place an order up to a cut-off time, some hours before delivery will take place. We assume that the likelihood that a customer places an order is independent of the delivery time slots offered and is not correlated to the order placement of other customers. When placing an order, a customer must select a delivery time slot during which delivery will take place at his delivery location. A vehicle that arrives early at a delivery location must wait. The retailer seeks to *design* a set of delivery routes, such that each customer, i.e., its delivery location, is visited on at least one of the routes, and associated time slots, one for each location visited, so as to maximize the *expected* revenue. We assume that the set of possible time slots that can be assigned to a customer location has already been decided. The time slots may overlap, but they all have the same width, and they cover the entire planning horizon. The subset of possible time slots for a delivery location contains those time slots that are feasible for that location, i.e., that overlap with the time period defined by the earliest time a vehicle can reach the location and the latest time a vehicle can depart the location to return to the fulfillment center before the end of the planning horizon.

As will become evident soon, even solving the special case in which the retailer has only a single vehicle with infinite capacity and assigns only a single time slot to each delivery location is surprisingly challenging and gives rise to insightful observations. For the remainder, therefore, we focus on this special case, leaving the general case for future research.

2 The single-vehicle case

The problem is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, with $\mathcal{V} = \mathcal{V}_c \cup \{o, d\}$ the set of vertices, where $\mathcal{V}_c = \{1, 2, \ldots, n\}$ is the set of customer delivery locations and where, for convenience, we represent the fulfillment center with a start and an end node, o and d, respectively, to be able to distinguish the departure and return of the vehicle, and with $\mathcal{A}$ the set of (directed) arcs connecting the nodes. We let $t_{ij} \geq 0$ denote the travel time associated with arc $(i, j) \in \mathcal{A}$. We assume that service times are included in the travel times, and that travel times satisfy the triangle inequality. The retailer has a single vehicle with unlimited capacity to make deliveries, which reflects that it is time rather than capacity that restricts the delivery route. When a time slot $s = [a_s, b_s]$ is assigned to a location in the delivery route, the earliest time a delivery can be made at that location is a_s and the latest time a delivery can be made at that location is b_s. A vehicle arriving early must wait at the location. A set $\mathcal{T}$ of possible time slots to be assigned to delivery locations is given. The time slots in $\mathcal{T}$ may overlap, but we assume their width is equal, and they cover the entire planning horizon $[0, T]$, with T the planning horizon. The set of possible time slots $\mathcal{T}_i \subset \mathcal{T}$ for location i contains the time slots which overlap with the period defined by the earliest time a vehicle reach that location, i.e., $t_{o,i}$, and the latest time a vehicle has to depart from that location (to return to the fulfillment center before the end of the planning horizon), i.e., $T - t_{i,d}$. The fulfillment center has time window $[a_o, b_o] = [a_d, b_d] = [0, T]$. We identify the set of customers $\mathcal{C}$ with their delivery locations, i.e., $\mathcal{V}_c$ (and use these interchangeably from now on). Each customer $c \in \mathcal{C}$ has an order placement probability $p_c \in (0, 1]$, and, when served, results in a revenue r_c for the retailer. We assume that order placement probabilities are iid and independent of the time slot assigned to the delivery location.

The retailer seeks to design an *a priori* delivery route, visiting all delivery locations, and associated time slots, one for each location, so as to maximize the *expected* revenue. Let Ω be the set of all possible scenarios of order placements. A single scenario $\omega \in \Omega$ can be described by a sequence of delivery locations, representing which customers have placed an order and in what sequence – the exact times of the order placements are not important. (Note that when the order placement probabilities are equal, i.e., $p_c = p$ for $c \in \mathcal{C}$, the possible scenarios are equally likely – given the iid assumption.) The revenue for a scenario $\omega \in \Omega$ is determined as follows. During the order placement phase, an arriving order is inserted in the *actual* delivery route, i.e., the delivery route to be executed after the cut-off time, based on the delivery location's position in the a priori route. That is, the delivery location is inserted in the actual delivery route after the delivery locations of orders placed earlier and that precede it in the a priori route, and before the delivery locations of orders placed earlier and that succeed it in the a priori route. After the insertion of an order, any delivery location that has become time infeasible, i.e., for which it is no longer possible to make a delivery during its assigned time slot, is removed, and orders for these locations will be skipped from that point on. After all orders in ω have been processed, i.e., have either been inserted or skipped, the revenue of the scenario is simply the sum of the revenues of the orders that have been inserted in the actual delivery route. The expected revenue for an a priori route is the sum of the revenues of all possible scenarios for that a priori route weighted by the probability of occurrence of the scenarios. Observe that (to keep operations simple) the delivery locations in the actual delivery route are visited in the same order as in the a priori route.

The basic problem can be stated formally as a stochastic optimization problem as follows:

$$\max_{\rho, y} \mathbb{E}_\omega(r(\rho, y, \omega)),$$

with $r(\rho, y, \omega)$ denoting the revenue of having a priori route ρ and time slot assignment y and customer order realization $\omega \in \Omega$. Note a customer order realization has two components: (1) the set of customers that want to place an order, and (2) the sequence in which these customers will place their order. Note too that some customer that want to place an order might be unable to do so because of time slots and planning horizon constraints given the set of customers that have already been accepted earlier.

Given our assumptions on the customer probabilities, we can write

$$\begin{aligned}\mathbb{E}_\omega(r(\rho, y, \omega)) &= \sum_{S \subset \mathcal{V}_c} p_S \sum_{\bar{\omega} \in \text{Perm}(S)} p_{\bar{\omega}} r(\rho, y, \bar{\omega}) \\ &= \sum_{S \subset \mathcal{V}_c} \prod_{i \in S} p_i \prod_{j \in \mathcal{V}_c \setminus S} (1 - p_j) \sum_{\bar{\omega} \in \text{Perm}\,(S)} \frac{1}{|S|!} r(\rho, y, \bar{\omega}),\end{aligned}$$

with $\mathcal{V}_c$ the set of customer locations, S the set of customers willing to placing an order, and $\text{Perm}\,(S)$ denoting the set of all permutations of set S, indicating all possible sequences in which the customers in set S place their order.

This can be viewed as a 2-stage stochastic program, where the a priori route ρ and time slot assignment y are first stage decision, and the evaluation is the second stage "decision". In the second stage, there are no recourse options, so it is purely an evaluation stage. By viewing the problem as a 2-stage stochastic program, it is natural to consider solution approaches for 2-stage stochastic programs. We have chosen to pursue a *sample average approximation* algorithm.

3 Sample Average Approximation

As observed earlier, we can view the problem as a 2-stage stochastic program without recourse decisions in the second stage, only first stage decision evaluation in the second stage. We have implemented two variants of a sample average approximation (SAA) algorithm, one in which the evaluation of first stage decision is sampling-based (as in the standard version of SAA), and one in which the evaluation of first stage decision is done exactly, using a specialized dynamic programming algorithm. The latter can be used for instances with up to 10 customer locations.

Even though we have shown above that it is not necessarily optimal for the a priori route to be an optimal TSP tour through the customer locations, fixing the a priori route (to the optimal TSP tour) simplifies the problem significantly as it reduce the problem to finding an optimal time slot assignment. Therefore, in the preliminary computational results presented next, we consider both the general and the restricted problem (with a priori route set to the TSP tour).

4 Computational Experiments

Instances with the following characteristics have been generated: depot location in centered or corner of the region, a few large, many small, non-overlapping, or overlapping time slots, a small, medium, and large planning horizon, sparse or dense sets of customer locations, and uniform or clustered customer locations. Results for some of these instances can be found in Tables 1 and 2. Here, n denotes the number of customers, w the time slot width and T the planning horizon length, the latter two both as fraction of the optimal TSP tour duration (i.e., with value 1.0 the TSP tour duration). As expected, we see that using larger sample sizes results in better solutions, i.e., with higher expected revenues, but also in higher computation times. Interestingly, high-quality solutions can already be achieved with relatively small sample sizes, e.g., $N = 8$.

Table 1: Some results for the SAA method with runs of $M = 20$ samples. Each row shows averages over 10 instances with $n = 4$ customers and centralized depot.

				Exact Average Revenue						Estimated Gap						Gap Std Dev ($\alpha = 0.05$)						Running CPU (s)					
	n	w	T	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$
c60	4	0.25	0.90	1.71	1.76	1.79	1.79	1.79	1.79	0.14	0.05	-0.04	-0.02	-0.01	-0.00	0.26	0.18	0.14	0.09	0.07	0.05	4.04	1.30	2.48	6.28	22.54	83.51
		0.25	0.75	1.54	1.56	1.56	1.56	1.56	1.56	0.07	0.05	-0.00	0.03	0.03	0.01	0.23	0.16	0.11	0.08	0.05	0.04	10.87	0.93	1.87	4.77	16.28	55.99
		0.25	0.60	1.30	1.30	1.31	1.31	1.31	1.31	-0.01	-0.03	0.00	0.01	0.01	-0.00	0.19	0.12	0.09	0.07	0.05	0.03	0.42	0.64	1.19	2.63	6.31	15.63
c60	4	0.125	0.90	1.71	1.72	1.73	1.73	1.73	1.73	0.07	0.09	0.05	0.02	0.03	0.02	0.27	0.18	0.12	0.09	0.06	0.05	17.80	1.67	2.95	9.56	45.33	290.87
		0.125	0.75	1.51	1.54	1.55	1.55	1.55	1.55	0.14	0.08	0.05	-0.01	0.01	0.02	0.23	0.15	0.11	0.08	0.06	0.04	0.56	0.87	1.78	4.35	13.75	85.83
		0.125	0.60	1.26	1.30	1.30	1.30	1.30	1.30	0.06	-0.00	0.01	0.00	-0.00	0.00	0.18	0.14	0.10	0.06	0.05	0.03	0.55	0.74	1.11	2.05	4.84	18.50

Table 2: Some results of the SAA method on the restricted problem of a fixed TSP route and with runs of $M = 20$ samples. Each row shows averages over 10 instances with n customers and centralized depot.

				Exact Average Revenue						Estimated Gap						Gap Std Dev ($\alpha = 0.05$)						Running CPU (s)					
	n	w	T	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$	$N=64$
c60	4	0.25	0.90	1.74	1.75	1.75	1.75	1.75	1.75	0.10	0.05	-0.02	-0.00	0.00	0.00	0.26	0.18	0.14	0.09	0.07	0.04	0.51	0.51	0.88	1.53	2.87	5.78
		0.25	0.75	1.54	1.54	1.54	1.54	1.54	1.54	0.07	0.05	-0.00	0.02	0.03	0.00	0.23	0.16	0.11	0.08	0.05	0.04	0.58	0.49	0.84	1.52	2.83	5.55
		0.25	0.60	1.28	1.28	1.28	1.28	1.28	1.28	0.01	-0.01	0.02	0.02	0.02	0.00	0.18	0.12	0.08	0.07	0.05	0.03	0.30	0.44	0.78	1.39	2.55	5.07
c60	4	0.125	0.90	1.70	1.70	1.70	1.70	1.70	1.70	0.07	0.08	0.04	0.02	0.03	0.01	0.26	0.17	0.12	0.09	0.06	0.04	1.51	0.62	1.11	1.97	4.51	15.85
		0.125	0.75	1.51	1.53	1.53	1.53	1.53	1.53	0.13	0.06	0.03	-0.03	0.00	0.00	0.22	0.15	0.11	0.09	0.06	0.04	0.33	0.56	0.99	1.73	3.30	12.00
		0.125	0.60	1.27	1.27	1.27	1.27	1.27	1.27	0.05	0.02	0.02	0.01	0.01	0.01	0.18	0.14	0.10	0.06	0.04	0.03	0.31	0.49	0.85	1.46	2.79	5.96
c60	8	0.25	0.90	3.57	3.70	3.70	3.70	3.70	3.70	0.27	0.07	0.03	0.01	0.01	0.02	0.42	0.27	0.19	0.13	0.09	0.07	2.41	3.10	5.05	10.44	28.87	142.09
		0.25	0.75	3.19	3.20	3.22	3.22	3.22	3.22	0.21	0.12	0.06	0.04	0.04	0.02	0.36	0.23	0.17	0.11	0.07	0.06	2.59	3.64	6.06	14.20	43.30	232.32
		0.25	0.60	2.57	2.60	2.61	2.61	2.61	2.61	0.19	0.15	0.08	0.06	0.03	0.01	0.30	0.19	0.15	0.10	0.07	0.05	2.82	3.54	5.20	8.64	16.44	40.02
c60	8	0.125	0.90	3.59	3.62	3.63	3.63	3.63	3.63	0.20	0.12	0.16	0.03	0.03	0.03	0.39	0.28	0.18	0.14	0.09	0.06	2.43	3.42	6.75	14.94	64.80	667.15
		0.125	0.75	3.06	3.16	3.18	3.18	3.18	3.18	0.32	0.19	0.12	0.06	0.04	0.03	0.32	0.23	0.18	0.11	0.08	0.06	2.69	3.87	7.10	17.29	59.38	650.39
		0.125	0.60	2.53	2.55	2.59	2.59	2.59	2.59	0.29	0.20	0.09	0.03	0.04	0.02	0.28	0.21	0.12	0.10	0.07	0.05	2.83	3.76	5.88	11.25	31.05	197.40

Acknowledgments

This research is funded by NWO, the Netherlands Organisation for Scientific Research, as part of the multi-annual research programme on Sustainable Logistics, and by Stichting Erasmus Trustfonds.

A column generation procedure for the Flexible Ship Loading Problem

Jonas Christensen
Department of Management Engineering
Technical University of Denmark, Kgs. Lyngby, Denmark
Email: jomc@dtu.dk

Dario Pacino
Department of Management Engineering
Technical University of Denmark, Kgs. Lyngby, Denmark
Email: darpa@dtu.dk

1 Introduction

In search of economies of scale container shipping lines are building bigger and bigger vessels. Over the last decade the average capacity of container ships has doubled, and as of May 2017, OOCL Hong Kong holds the world record for the largest containership, with a carrying capacity at 21,413 TEU. That is a factor 2.6 increase compared with the ∼8,200 TEU record set in 2003. It is expected that both the average and maximum size of containerships will grow over the coming years.

For the carriers, these new mega-vessels provide large unit cost savings compared to older and smaller vessels. However, a significant amount of the costs savings is attributed to the emergence of slow steaming. Comparing modern mega-vessels with modern smaller vessels the cost savings are significantly lower, and the benefits of the mega-vessels are thus set to diminish when older medium-sized vessels are decommissioned. [3]

On the container terminal side, bigger vessels require more crane moves, and terminals are under pressure to minimise turnaround times. Minimizing the turnaround time makes it possible for the carriers to realise more of the savings potential that comes with the bigger vessels, as they will not have to speed up to stay on schedule due to port delays. For the terminal, improving productivity and minimising turnaround times helps to free up berth space, and clears up capacity for other vessels. [2]

While mega-vessels may cut unit costs for carriers, the total system costs are not reduced. Additional costs for ports, insurance companies and transport providers lead to higher total system costs as vessel sizes grow. Also, the general network structure leads to more transhipments and fewer direct services. Building even bigger vessels is therefore not a viable solution to deal with the diminishing benefits of the mega-vessels. Instead, the industry must improve operational efficiency. [4]

Acknowledging that improving terminal productivity is a shared goal between the carrier and the terminal, the Flexible Ship Loading Problem (FSLP) investigates a collaboration between the terminal and liner shipping companies. The liner provides the terminal with a stowage plan based on container classes. The terminal then has the flexibility of determining the position of the specific containers, as long as it adheres to the provided stowage plan. The terminal will assign containers to specific slots on the vessel, while also scheduling transfer vehicles to retrieve the container from the yard and deliver it in front of the crane. Doing so the terminal can better plan what container to be loaded at which time, thus giving the terminal better conditions to minimise the turnaround time for the vessel. The terminal also benefits from this collaboration as they can plan the use of their container-handling equipment better.

The FSLP was first introduced in [1]. We extend this work and formulate a generalised set cover model for the FSLP. The pricing problem is solved using multiple heuristics, if they all fail, an exact MIP model is used. The new mathematical formulation is shown to provide substantially better lower bounds.

2 The Flexible Ship Loading Problem

In the FSLP a liner vessel docked at a port is considered. The containers destined for the port have been unloaded, and a set of containers are to be loaded on the ship. The liner provides the terminal with a class-based stowage plan. A class-based stowage plan specifies that a container of a specific *class* is to be loaded at a given position of the vessel. Here, a container class corresponds to the dimensions of the container, properties (reefer or dry cargo container), destination and weight of the container (e.g. light, medium or heavy). The exact weight of the container might not have an impact on the feasibility of the stowage plan, and thus it is sufficient to consider weight classes. The terminal is responsible for loading the vessel, following the stowage plan. The class-based stowage plan leaves much freedom for the terminal which they wish to exploit, to optimise their operations. The terminal might have multiple containers of the same type, and thus they want to determine which container goes where on the vessel, to optimise their workload while ensuring the vessel leaves as planned.

Consider a set of containers ($\mathcal{C}$) to be loaded in positions on the ship ($\mathcal{P}$) by a set of Quay Cranes

(QCs) (the set $\mathcal{Q}$). The FSLP covers the assignment and scheduling of Transfer Vehicles (TVs) (set $\mathcal{S}$) to retrieve the containers from the yard and deliver in front of the Quay Crane (QC). We assume that the loading order for each QC is determined beforehand, and is known. The crane loading time is β, and thus there must be at least β time units between the loading of two successive positions.

The contract between the terminal and the liner specifies that with the amount of containers to be unloaded, the terminal is expected to finish loading the vessel at a given time - the Expected Finishing Time (EFT). If this is not ensured the terminal must pay a penalty of γ for every time unit the vessel is delayed. The terminal must pay the operators operating the TVs, for the time they work. The time unit cost of this is α, and we assume this cost also includes equipment wear and tear and maintenance. The terminal aims to minimise the sum of these two costs.

Let Ω_q be the set of all possible assignments for QC $q \in \mathcal{Q}$. With this, let the variable $y_{qa} \in \{0,1\}$ be a binary variable denoting if the QC $q \in \mathcal{Q}$ uses assignment $a \in \mathcal{Q}_q$. For the assignments, we define b_{ac} as 1 if assignment a assigns container c to a position, and 0 otherwise. Moreover, we define w_a as the service length for assignment a, and $\bar{z}_a$ as the finish time for assignment a. Lastly, let the variable ΔEFT be the maximum tardiness of the operations. With this, we can model the Flexible ship loading problem as seen below.

$$\texttt{Min Z} = \alpha \sum_{q \in \mathcal{Q}} \sum_{a \in \Omega_q} w_a y_{qa} + \gamma \Delta EFT \tag{1}$$

Subject to:

$$\sum_{a \in \Omega_q} y_{qa} = 1 \qquad \forall q \in \mathcal{Q} \tag{2}$$

$$\sum_{q \in \mathcal{Q}} \sum_{a \in \Omega_q} b_{ac} y_{qa} = 1 \qquad \forall c \in \mathcal{C} \tag{3}$$

$$\Delta EFT \geq \sum_{a \in \Omega_q} \bar{z}_a y_{qa} - EFT \qquad \forall q \in \mathcal{Q} \tag{4}$$

$$y_{qa} \in \{0,1\} \qquad \forall q \in \mathcal{Q}, a \in \Omega_q \tag{5}$$

$$\Delta EFT \geq 0 \tag{6}$$

The objective (1) function minimises the previous described cost. Constraint (2) ensure that every crane is assigned exactly one assignment, and constraint (3) make sure that every container is assigned to a position. The value of the variable ΔEFT is set in constraint (4), and (5) and (6) defines the variables. We denote this model (constraint (1)-(6)) as the Master Problem (MP)

3 Solution Method & Preliminary Results

The model MP contains an exponential number of variables, and generating all the feasible assignments is only possible for the smallest toy instances. Instead we will consider only a small subset of

all assignments, $\widehat{\Omega}_q$, and generate more assignments as they are needed. Using column generation we can then prove LP-Optimality of the Master Problem.

Finding the variables to add to the RMP, entails solving a pricing problem. Here we aim to find the variable with the most negative reduced cost, while ensuring the assignment is feasible for the set $\widehat{\Omega}_q$. When there doesn't exists any more negative reduced cost columns we have proved that the current RMP solution is an LP-optimal solution for the MP.

Two pricing problem heuristics are used to find new variables to add. If both of these fail to find any negative reduced cost variables, an exact MIP model is used. Also, a primal heuristic is used to get feasible solutions.

Table 1 shows preliminary results for the column generation method and compares with a compact formulation of the problem. Without branching, the column generation finds and proves Integer optimality of 4 out of 6 instances. This is attributed to the primal heuristic. The exact pricing method is by far the most time-consuming part of the column generation method.

				Compact-FSLP				**CG-FSLP**						
$\|C\|$	$\|CT\|$	$\|Q\|$	D	x^{Root}	x^{LB}	x^{UB}	$t(s)$	x^{LB}	x^{UB}	t^{Primal}_{Heu}	$t^{Pricing}_{Heu}$	t^{Master}	$t^{Pricing}_{Exact}$	$t(s)$
60	10	2	*LD*	1530	1544	1665	10800†	**1665**	**1665**	0.5	76.7	3.5	17528.3	17609.0
60	10	2	*S*	1020	1020	1020	116	**1020**	**1020**	0.8	34.5	1.4	0.1	36.8
60	10	2	*U*	1657	1670	1790	10800†	1696.7	1710	5.7	948.2	22.9	17029.7	18006.6†
60	25	2	*LD*	2060	2234	2275	10800†	**2250**	**2250**	0.2	92.5	0.1	194.6	287.5
60	25	2	*S*	1360	1360	1430	10800†	1365	1370	1.7	565.6	7.5	15638.9	16213.6
60	25	2	*U*	1490	1490	1490	721	**1490**	**1490**	0.6	74.6	0.3	0.1	75.6

Table 1: Preliminary Computational Results. † means the computation was terminated due to a timelimit

References

[1] Çagatay Iris, Jonas Christensen, Dario Pacino, and Stefan Ropke. Flexible ship loading problem with transfer vehicle assignment and scheduling. *Transportation Research Part B: Methodological*, 111:113 – 134, 2018.

[2] JOC. Berth productivity: The trends, outlook and market forces impacting ship turnaround times. In *JOC Port Productivity*. JOC Group, July 2014.

[3] OECD/ITF. The impact of mega-ships. In *Case-Specific Policy Analysis*. OECD/ITF, April 2015.

[4] UNCTAD. Review of maritime transport 2016. In *United Nations Conference on Trade and Development*, 2016.

A Survivable p-Hub Median Problem and a Modified Benders Decomposition Method

Hamid Mokhtar
School of IT and Electrical Engineering,
The University of Queensland, St Lucia QLD 4072, Australia
Email: h.mokhtar@uq.edu.au

Mohan Krishnamoorthy
School of IT & Electrical Engineering,
The University of Queensland, St Lucia QLD 4072, Australia

Andreas T. Ernst
School of Math. Sciences, Monash University, Clayton VIC 3800

1 Introduction

Hubs are employed in several network design contexts that involve flow interchange between nodes and are often used in the design of, for example, airline networks, parcel delivery networks, and telecommunication networks. Flow between nodes (referred to as *access nodes*) is routed via *hubs*, each of which acts as a consolidator and forwarder. Hub location problems (HLP) are very important class of problems in transportation and communication networks. Despite all the attention in the literature on the study of HLPs, there has not been a significant amount of attention paid to hub network survivability. This requirement is particularly relevant in electrical and telecommunication networks that have a hub topology.

In this paper, we introduce a new problem, namely the uncapacitated 2-allocation p-hub median problem (U2ApHMP) which is a modification well-known hub median problem which enables us to generate a hub network that is able to survive *access link* failures. Survivability is a feature of the U2ApHMP design and is an attempt to avoid severe costs of network disruptions. Then, we develop an improved Benders decomposition method for the U2ApHMP. In our approach, we address slow convergence of the method by some 'core point'. We take advantage of these improved cuts and enhance this approach by choosing better core points and generating stronger cuts. We also come up with a more efficient approach for solving subproblems to generate cuts.

2 Problem Statement

We consider a complete digraph (N, A), where A is the set of all arcs, $N = \{1, 2, \ldots, n\}$ is the set of nodes. Hubs are connected through a complete graph on the set of hubs, and non-hub nodes are only connected to hubs. We suppose the triangular inequality holds. So, all flow must be routed through at most two hubs. Thus, any path between i and j must contain three links, (i, k), (k, l), and (l, j), where i and j are connected to hubs k and l respectively. In practice, the cost of flow between different types of nodes has different cost coefficients: the *collection* coefficient χ corresponds to flow from a non-hub to a hub, the *distribution* coefficient α corresponds to flow from a hub to a non-hub, and the *transfer* coefficient δ corresponds to flow between hubs. Usually $\alpha \leq 1$, $\chi \geq \alpha$ and $\delta \geq \alpha$. The U2ApHMP is the problem of locating p hubs in N and allocating each non-hub to exactly 2 hubs with minimum total cost of fulfilling flow demands.

Theorem 2.1. U2ApHMP *is NP-hard, even when the location of hubs are fixed.*

3 Benders Decomposition

Benders decomposition method is a partitioning algorithm applied to mixed integer programming and nonlinear integer programming problems [1]. As an advantage, larger instances of problems can be solved since master problem (MP) and subproblem (SP) are often more tractable than the original problem. In each iteration of this method for U2ApHMP, the variables corresponding to the set of hubs and connection of non-hubs to hubs are fixed. The SP problem is a routing problem for n^2 pairs of nodes, where the underlying network is defined by the fixed variables in the MP. There are two issues with this decomposition: (i) slow convergence and a large number of iterations, and (ii) the computational effort to solve n^2 subproblems is very expensive. To tackle these issues, we first develop a modification of Benders decomposition. For the second issue, we model subproblems as minimum cost network flow problems to solve them more efficiently.

3.1 Accelerating the Benders Decomposition Approach

The optimal solution of SPs is not unique since the SP is degenerate. As a result, Benders cuts exist for the MP, with different strengths. The strength of Benders cuts is dependent on the choice of optimal solutions of SPs. Magnanti and Wong [4] proposed an acceleration of the Benders method, in which a second LP is constructed from the dual of the subproblem to maximise a weighted summation of the dual variables among optimal solutions. Let m_{ik}, m_{jl} for $k, l \in N$ be non-negative real parameters and m_0 be a real parameter. For $(m_0, \boldsymbol{m}_{ij}) = (m_0, m_{i1}, \ldots, m_{in}, m_{j1}, \ldots, m_{jn})$, we consider the following function as the objective function of SPs to generate cuts:

$$m_0 f_{ij} - \sum_{k \in N} m_{ik} u_{ijk} - \sum_{l \in N} m_{jl} v_{ijl} \tag{1}$$

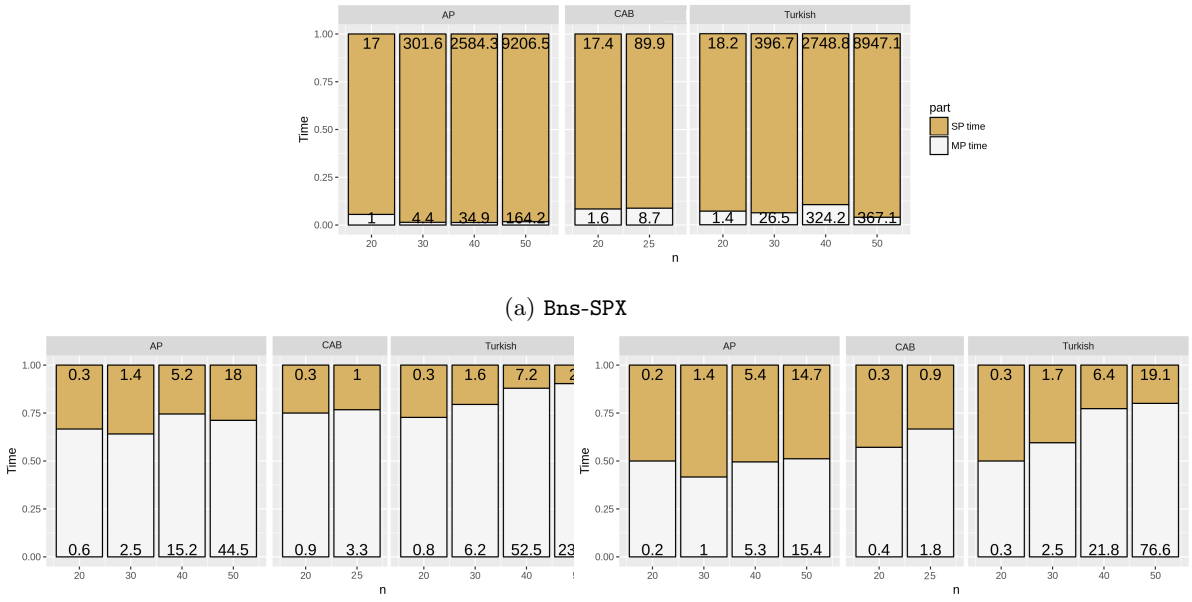
In experiments, we find an appropriate $(m_0, \boldsymbol{m})$ for stronger cuts and fewer required iterations [5].

3.2 Solving Subproblems BDS_{ij} Efficiently

Instead of using the simplex method to solve n^2 SPs in each iterations, we convert the SPs into a network routing problem and use the minimum cost network flow problem to obtain Benders cuts more efficiently. In our approach, we (i) avoid numerical instability, and (ii) solve the subproblems much more efficiently, so that generating n^2 cuts is not a hindrance to obtaining tight cuts for the MP formulation, as already observed in [2].

4 Computational Results

Our computational experiments were carried out on three well-known datasets in the HLP: the Civil Aeronautics Board (CAB) dataset [6], the Australia Post dataset (AP) [3], and Turkish Cargo Delivery dataset [7]. We observe that the modified Benders decomposition method is very efficient for solving U2ApHMP, and that our choice of core points significantly improves the convergence rate. Additionally, it also reduces the number of iterations required.



(a) `Bns-SPX`

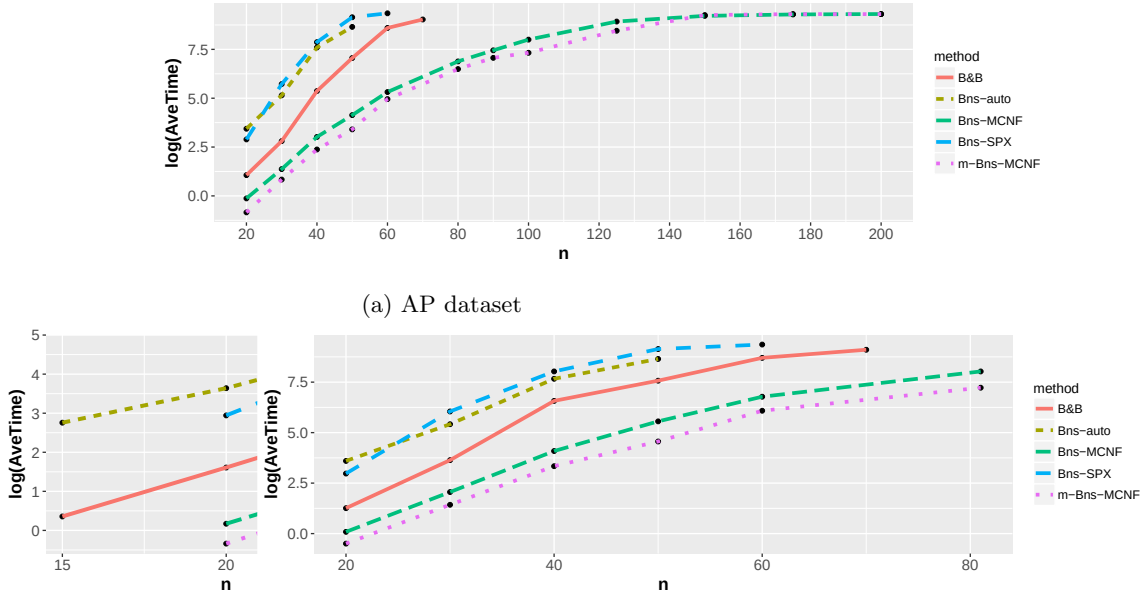
(b) `Bns-MCNF`

(c) $\boldsymbol{m}$-`Bns-MCNF`

Figure 1: The portion of computational efforts for solving SPs and MP for $20 \leq n \leq 50$ by methods

We perform experiments with three methods: (i) `Bns-SPX` which is the modified Benders decomposition, where n^2 SPs are solved using the simplex method, (ii) `Bns-MCNF` which is the modified Benders decomposition, where SPs are solved using improved solution approach, and (iii) $\boldsymbol{m}$-`Bns-MCNF` which is `Bns-MCNF` where the objection function is defined by (1). As Figure 1 shows, our

approach, not only reduced the computational burden of solving SPs, but also required less computational efforts for MPs on average, due to generation of stronger cuts. The modified Benders decomposition together with efficient method of solving subproblems significantly improves our ability to tackle large instances of U2ApHMP. Note that $\boldsymbol{m}$-`Bns-MCNF` outperforms `Bns-MCNF` in average computational effort or in the best solution gap for large instances. Also, as Figure 2 shows, our approach outperforms any other approach, including branch and bound (B&B) and the built-in Benders method in Cplex (Bns-Auto).



(a) AP dataset

(b) CAB dataset

(c) Turkish dataset

Figure 2: Comparison of average solution times on U2ApHMP with respect to the number of nodes

References

[1] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4(1):238–252, 1962.

[2] R.S. de Camargo, G.D. Miranda, and H. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Comput. Oper. Res.*, 35(4):1047–1064, 2008.

[3] A.T. Ernst and M. Krishnamoorthy. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, 4(3):139–154, 1996.

[4] T.L. Magnanti and R.T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.*, 29(3):464–484, 1981.

[5] Hamid Mokhtar, Mohan Krishnamoorthy, and Andreas T. Ernst. The 2-allocation p-hub median problem and a modified benders decomposition method for solving hub location problems. *Computers Operations Research*, 2018.

[6] M.E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *Eur J Oper Res*, 32(3):393–404, 1987.

[7] P.Z. Tan and B.Y. Kara. A hub covering model for cargo delivery systems. *Networks*, 49(1):28–39, 2007.

The pickup and delivery problem with on-line transfers, for the next generation of public transport.

Paul Bouman

Econometric Institute, Erasmus School of Economics

Erasmus University Rotterdam, The Netherlands

Gizem Özbaygın

Faculty of Engineering and Natural Sciences

Sabancı University, Istanbul, Turkey

Lucas P. Veelenturf

School of Industrial Engineering

Eindhoven Univeristy of Technology, The Netherlands

Email: l.p.veelenturf@tue.nl

October 14, 2018

1 Introduction

We introduce and study a new variant of the vehicle routing problem, which we call the pickup and delivery problem with online transfers (PDPOT), motivated by an innovative passenger transportation concept involving self-driving vehicles (see e.g. `http://www.next-future-mobility.com/`). These vehicles are designed in a way that they can couple/decouple while en-route and transfer passengers seamlessly towards more efficient capacity utilization and traffic management. Due to the potential reduction in fuel/energy consumption and travel costs, there are studies in the vehicle routing literature taking transfer opportunities into account within their framework. The most closely related vehicle routing problem to the one we consider in this study is the pick-up and delivery problem with transfers (see e.g. [1]). However, the main difference and perhaps the most challenging aspect of the PDPOT is that when two or more vehicles couple, the passengers may transfer from one vehicle to another during the time the vehicles are traveling together as a single

vehicle. Among the major contributions of our study are: (1) the development of an optimization based approach to solve a complex vehicle routing problem arising in an on-demand transportation system involving autonomous shared vehicles, and (2) we aim to analyze the benefits of online transfers in such an environment.

2 Problem Definition

Now we formally introduce the pickup and delivery problem with online transfers (PDPOT). Consider a road network defined by the following sets of types of locations: P: locations with a parking lot (i.e. where vehicles are allowed to wait), S: location with a short stop possibility (locations where the vehicles are allowed to stop shortly, and possibly pick up/drop-off passengers), and I: all other road junctions (locations where stops are not allowed). This means that the total set of locations is $R = P \cup S \cup I$. Each location has at least one link (i.e. road) to another location. And each location is reachable from another location by a number of consecutive links. The set of links is denoted by $E \subseteq R \times R$ and for each link $(i, j) \in E$ we have a fixed travel time $\tau_{i,j}$.

Let K and V denote a set of passengers and the set of vehicles, respectively. Each passenger $k \in K$ is associated with an origin $o_k \in O_K$ at which he wants to be picked up after e_k, and a destination $d_k \in D_K$ which he wants to reach before l_k. Similarly, each vehicle $v \in V$ has an origin $o_v \in O_V$ where it starts service at e_v and a destination $d_v \in D_V$ where the vehicle should end its route at l_v. We assume that the fleet consists of identical vehicles, each having a capacity for Q passengers. All origins and destinations of passengers and vehicles are among the locations in R (i.e., $O_K \subseteq R$, $D_K \subseteq R$, $O_V \subseteq R$ and $D_V \subseteq R$). To formulate this problem, we discretize time into a set T of time points and define a time-expanded version of the original road network.

The aim is to find a set of routes for the homogeneous fleet of vehicles that will transport all passengers from their origins to their destinations within their respective time windows at minimum total cost, which is determined mainly by two components: (1) total energy consumption of the vehicles, (2) total convenience for the passengers. We assume that when two or more vehicles couple and travel together as a single vehicle, the energy consumption is reduced compared to the case where they travel separately. Besides saving energy, this coupling and decoupling mechanism allows for online passenger transfers, which may also lead to additional savings due to the potential decrease in the overall travel distance.

We consider two types of transfers: online and offline, the former corresponding to the transfers that take place while the vehicles are coupled, whereas the latter refers to the transfers in which a passenger is dropped off by a vehicle and later picked up by another one (possibly after waiting for a while). We assume that the vehicles are not allowed to (1) stop at road intersections, and (2) wait longer than one unit of time at stops.

As mentioned earlier, the objective function includes routing cost and passenger (in)convenience.

Routing cost: For traversing an arc a the travel cost for a vehicle is c_a^d. However, if multiple vehicles traverse the arc a in a platoon, only the first vehicle has travel cost c_a^d, and all other connected vehicles have travel cost $(1-\eta)c_a^d$. This means that if $n > 0$ vehicles traverse the arc a, the total travel cost is $nc_a^d - \eta(n-1)c_a^d$.

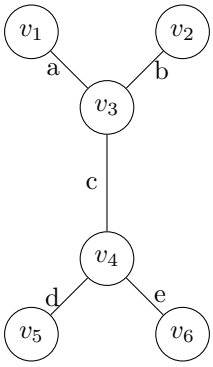
Passenger (in)convenience: We assume that each passenger wants to reach his destination as soon as possible with as few outside transfers as possible. Therefore we introduce a reward c^{early} for each minute a passenger arrives earlier than planned, and a fixed penalty c^{out} per outside transfer a passenger has to make (i.e. inside transfers are not penalized). Note that it may not be possible to serve all passenger requests. Therefore, we also introduce a fixed penalty c^{reject} for each rejected passenger.

3 Methodology

Our initial goal is to develop an efficient solution methodology for the PDPOT and test it on instances of reasonable size. Later, we may also work on different but related problems (like the dynamic variant). First we proposed an arc-based and a path-based formulation for the problem which is an extension of a multi-commodity network flow model. For the path-based formulation, we developed a branch-price-and-cut methodology. At each pricing iteration, vehicle and/or passenger paths are generated, and in the restricted master problem (RMP), the paths are linked such that passengers can only follow a path if there are vehicles traversing each arc on this path and if the transfer restrictions are met. Some important issues in tuning the methodology are the following: (i) High symmetry in the vehicle paths. (ii) The large number of pricing problems. It is not trivial how many and of which type (vehicle or passenger pricing problems) to solve before updating the duals by solving the linear relaxation of the RMP again. For example, if the duals are not updated in between generating vehicle paths, a lot of similar columns will be generated for different vehicles. (iii) Developing a warm start feasible solution. To find a starting solution, we may make use of an existing methodology for the Pickup and Delivery Problem with Transfers like [1]. However, this requires some transformation processes as we are making use of a road network based time expanded graph.

4 Illustrative Example

To give an idea about the potential benefit that can be achieved using en-route transfer, we provide a small illustrative example in Figure 1. In this example we have two vehicles with at least two seats. The first vehicle departs from location v_1 and must end its operations at location v_5. The second vehicle departs from location v_2 and ends its operations at location v_6. We have four



(a) Example Network

		Time step			
	O.	1	2	3	D.
Veh. 1	v_1	a	c	d	v_5
Veh. 2	v_2	b	c	e	v_6
Pass. 1	v_1	a	c	d	v_5
Pass. 2	v_1	a	c	e	v_6
Pass. 3	v_2	b	c	d	v_6
Pass. 4	v_2	b	c	e	v_6

(b) En-route transfers on c

		Time step				
	O.	1	2	3	4	D.
Veh. 1	v_1	a	v_3	c	d	v_5
Veh. 2	v_2	b	v_3	c	e	v_6
Pass. 1	v_1	a	v_3	c	d	v_5
Pass. 2	v_1	a	v_3	c	e	v_6
Pass. 3	v_2	b	v_3	c	d	v_6
Pass. 4	v_2	b	v_3	c	e	v_6

(c) External transfers at v_3

Figure 1: Illustrative example network with two example schedules where the schedule with an en-route transfer at arc c is shorter than the schedule with an external transfer at location v_3.

passengers: two with location v_1 as their origin, and two with location v_2 as their origin. At both origins, one of the passengers has location v_5 as the destination while the other has v_6 as the destination. When en-route transfers are allowed, the two vehicles can couple at arc c and the passengers can make their transfer while the vehicle is moving. In the case where no en-route transfers are allowed, the two vehicle must make a stop at location v_3 (or v_4) so that the passengers can transfer. As the vehicles have to make an extra stop compared to the first case, it is clear that this solution takes more time than the solution with en-route transfers.

5 Final remarks

We address a new vehicle routing problem variant within the context of an on-demand public transportation system involving self-driving pods that can couple/decouple to facilitate online passenger transfers. The main motivation behind this work is to study a complex vehicle routing problem which is not only interesting from a theoretical (and a methodological) point of view, but can also provide useful insights into the next generation of public transport. To solve the PDPOT and to analyze the benefits of online transfers in a pickup and delivery setting, we propose and implement a branch-cut-and-price algorithm.

References

[1] R. Masson, F. Lehud, and O. Pton, " An adaptive large neighborhood search for the pickup and delivery problem with transfers", *Transportation Science* 47(3), 344-355 (2013).

Ship routing problem with berthing time clash avoidance constraints and minimizing demurrage

Kosuke Kawakami

Department of Industrial Engineering and Economics

Tokyo Institute of Technology,

Intelligent Algorithm Research Center

Nippon Steel

Email: kawakami.t9p.kohsuke@jp.nipponsteel.com

Mirai Tanaka[1]

Department of Statistical Inference and Mathematics

The Institute of Statistical Mathematics

1 Introduction

In Japanese steel industry, almost all raw materials such as coal and iron ore are imported via maritime transportation. As we spend huge costs on the maritime transportation, it is highly required to reduce the maritime transportation costs by solving ship routing problems.

While the mathematical structure of the ship routing problem is similar to that of a vehicle routing problem (VRP), there are some differences between the both problems. For example, in a VRP, vehicles such as trucks can be mass-produced. Thus, we can ignore the differences by individual vehicles. On the other hand, in a ship routing problem, we have to consider the different characteristics of individual vessels. Moreover, while vehicles in a VRP can park near the destinations almost anytime, vessels in a ship routing problem are impossible to berth while other vessels occupy the berth.

The first difference of the ship routing problems has been studied widely for recent two decades. Christiansen et al. [1] have provided a comprehensive survey on the topic. However, there is little published information on the second difference. Pang et al. [2] introduced a set partitioning modeling and a column generation approach to avoid berthing time clash in the ship routing problem. They defined the master problem that prohibits berthing time clash. Then, by heuristically solving the constraint shortest path problem (CSPP) constructed on the time-space network, they found a shipping plan considering berthing time clash avoidance.

In this paper, we consider a ship routing problem for the raw materials such as coal and iron ore. Comparing our model with the model of Pang et al. [2], we consider not only berthing time clash avoidance but also minimizing the demurrage. The demurrage means the detention of a vessel during loading or unloading beyond the schedule time of departure. When the demurrage occurs, we have to pay the extra fee in proportion to the time. Since the model of Pang et al. [2] targeted for small-size feeder vessels whose demurrage penalty is cheap, they did not have to consider the cost of demurrage. However, since we target for huge vessels with the weight of a load more than 200 kilotons whose

[1] The second author's research was supported by JSPS KAKENHI Grant Numbers 16K16357.

demurrage penalty is extremely high, we must take into account of minimizing the cost of demurrage. A key concept of this paper is that we contrive a new structure of the time-space network which naturally minimizes the demurrage by solving a constrained shortest path problem.

As another difference, because we target transport problems of raw materials such as ore and coal which are imported from distant loading ports, we assume that every vessel leaving the loading ports unloads all raw materials at the unloading ports.

Table 1: An example of order requests with multiple ports unloading

Order name	Loading port	Unloading port	Unload amount at each port
b_1	k_1	k_3	90
b_2	k_2	k_3, k_4	90, 120

Table 1 shows a simple example of the order requests considering with multiple ports unloading. As shown in the Table 1, the order request b_1 is a simple request which loads 90 kilotons of raw materials at the port k_1 and unloads 90 kilotons at the port k_3, whereas the order request b_2 is a more complex request, because it includes multiple ports unloading, which loads 210 kilotons and unloads 90 kilotons at the port k_3, 120 kilotons at the port k_4. In the Japanese steel industry, since the raw materials are transported from abroad, the distance between loading ports and unloading ports is far enough. For this reason, we assume that the vessels which depart from the loading port must unload all the raw materials at the unloading port before the next cruise which goes to a new loading port. Even if such the assumptions are imposed, it does not affect the optimality of the problems.

The problem described above can be modeled as a set partitioning problem having constraints on berthing clash avoidance and order requests. In the following sections, we describe the master problem and the pricing problem for the set partitioning problem.

2 Master problem

As described in Pang et al. [2], we divide the planning horizon $[0, T_{\max}]$ into discrete time intervals and assume that a vessel can only at any location at the beginning of the interval. On the set partitioning formulation for the scheduling problem, the master problem is described as follows:

$$\text{minimize} \quad \sum_{v \in V} \sum_{R \in \Omega_v} c_R^v X_R^v \tag{1}$$

$$\text{subject to} \quad \sum_{v \in V} \sum_{R \in \Omega_v} \delta_{bR}^v X_R^v = 1 \ (\forall b \in B), \tag{2}$$

$$\sum_{R \in \Omega_v} X_R^v = 1 \ \ (\forall v \in V), \tag{3}$$

$$\sum_{v \in V} \sum_{R \in \Omega_v} \sigma_{\rho g R}^v X_R^v \leq 1 \ \ (\forall \rho \in L, \forall g \in U), \tag{4}$$

$$X_R^v \in \{0,1\} \ (\forall v \in V, R \in \Omega_v), \tag{5}$$

where V is the set of vessels, Ω_v is the set of feasible routes for a vessel $v \in V$, c_R^v is the cost of a vessel v on the route $R \in \Omega_v$, X_R^v is a binary variable and equal to 1 if a vessel v selects the route $R \in \Omega_v$ otherwise 0, B is the set of orders, δ_{bR}^v is 1 if order $b \in B$ is served on route R by a vessel v, and δ_{bR}^v is 0 otherwise, L is the set of ports, U is the set of discrete time intervals obtained by discretizing $[0, T_{\max}]$, and $\sigma_{\rho g R}^v$ is 1 if a vessel $v \in V$ on route R occupies the port $\rho \in L$ at time period $g \in U$. Objective function (1) denotes the total travel cost of the vessels. Constraint (2) ensures that each order is handled only once. Constraint (3) ensures that each vessel must take exactly one route.

Constraint (4) states that each loading or unloading port is occupied by less than one vessel during any time interval. Constraint (5) defines the binary variables.

Since the master problem (1)–(5) has an exponential number of route columns, we use a column generation approach that is commonly used to solve the linear relaxation problem. As described in [2], the expression of the reduced cost $\bar{c}_R^v$ is as

$$\bar{c}_R^v = c_R^v - \sum_{b\in B} \pi_b' \delta_{bR}^v - \pi_v'' + \sum_{\rho\in L}\sum_{g\in U} \pi_{\rho g}''' \sigma_{\rho gR}^v, \tag{6}$$

where $\pi_b'(b \in B), \pi_v''(v \in V)$, and $\pi_{\rho t}''' \geq 0\ (\rho \in L, g \in U)$ are the dual variables associated with the constraints (2)–(4). Then, we can generate a new route for the master problem (1)–(5) by determining that the value $\min_{v\in V}\{\bar{c}_R^v : v \in V, R \in \Omega_v\}$ is negative. This problem can be decomposed into the constrained shortest path problems (CSPP) for individual vessels. In the following section, we discuss about a new structure of the CSPP.

3 Pricing problem

Let $\bar{G}_v = (\bar{N}_v, \bar{A}_v)$ be a directed network modeling a shortest path problem for each vessel $v \in V$. In order to represent the demurrage in the network, we define two types of node for each port indexed by $h \in \{-,\ +\}$. Nodes indexed by "$-$" represents the arrival time and the demurrage of the vessel and ones indexed by "$+$" represents the departure time of the vessel. Moreover, to express the multiple ports unloading, we define the index $k \in L$ which represents the current unloading port, and the set $K \subset L$ which represents the set of unloading ports that can be called. The set of nodes $\bar{N}_v$ is described as

$$\bar{N}_v = \{s_v, t_v\} \cup \left(\bigcup_{g\in U} \left\{ L_{b,h}^g : b \in B, h \in \{-,+\} \right\} \cup \left\{ D_{b,h}^{g,k,K} : b \in B, h \in \{-,+\}, k \in L, K \subset L \right\} \right),$$

where s_v is the source node, t_v is the sink node for each vessel $v \in V$, $L_{b,h}^g$ is the node corresponding to arrival or departure of the loading port of order b at time period g, and $D_{b,h}^{g,k,K}$ is the node corresponding to arrival or departure of the unloading port k of the callable port sets K of order b at time period g. Note that since the loading port and unloading port are completely separated in the steel industry, we represented them as the different nodes. Next, we define arcs on the nodes considering the constraints for the order requests, and port constraints. The set of arcs $\bar{A}_v$ is described as follows:

$$\bar{A}_v^1 = \left\{ \left(s_v, L_{b,-}^g\right) : b \in B;\ g = T_{s_v, L_{p,-}^g} \right\},$$
$$\bar{A}_v^2 = \left\{ \left(L_{b,-}^g, L_{b,-}^{g+1}\right) : b \in B; g \in U \right\},$$
$$\bar{A}_v^3 = \left\{ \left(L_{b,-}^f, L_{b,+}^g\right) : b \in B; f \in U;\ g = f + \tau_{L_{b,-}^f, L_{b,+}^g} \right\},$$
$$\bar{A}_v^4 = \left\{ \left(L_{b,+}^f, D_{b,-}^{g,k,K}\right) : b \in B; f \in U;\ g = f + T_{L_{b,+}^f, D_{b,-}^{g,k,K}} \right\},$$
$$\bar{A}_v^5 = \left\{ \left(D_{b,-}^{g,k,K}, D_{b,-}^{g+1,k,K}\right) : b \in B; g \in U \right\},$$
$$\bar{A}_v^6 = \left\{ \left(D_{b,-}^{f,k,K}, D_{b,+}^{g,k,K}\right) : b \in B; f \in U;\ g = f + \tau_{D_{b,-}^{f,k,K}, D_{b,+}^{g,k,K}} \right\},$$
$$\bar{A}_v^7 = \left\{ \left(D_{b,+}^{f,k,K}, D_{b,-}^{g,m,M}\right) : b \in B; f \in U;\ g = f + T_{D_{b,+}^{f,k,K}, D_{b,-}^{g,m,M}}; m \in L;\ M \subset L \right\},$$
$$\bar{A}_v^8 = \left\{ \left(D_{b,+}^{f,k,K}, L_{q,-}^g\right) : b \in B; f \in U;\ g = f + T_{D_{b,+}^{f,k,K}, L_{q,-}^g};\ q \in B \backslash \{b\} \right\},$$
$$\bar{A}_v^9 = \left\{ \left(D_{b,+}^{g,k,K}, t_v\right) : b \in B; g \in U \right\},$$
$$\bar{A}_v = \bar{A}_v^1 \cup \bar{A}_v^2 \cup \bar{A}_v^3 \cup \bar{A}_v^4 \cup \bar{A}_v^5 \cup \bar{A}_v^6 \cup \bar{A}_v^7 \cup \bar{A}_v^8 \cup \bar{A}_v^9,$$

where T_{ij} represents the discrete cruise time from a departure node i to an arrival node j, and τ_{ij} means the discrete loading or unloading time from an arrival node i to a departure node j. For $i, j \in \bar{N}_v$, arc $(i, j) \in \bar{A}_v$ means that a vessel can be moved from the node i to the node j. In order to match the objective value between the optimal solution of the CSPP and the shortest s_v-t_v path on the time-space network, we set $f(\cdot)$ as the cost function for each arcs of the time-space network defined as follows:

$$f(\bar{A}_v^1), f(\bar{A}_v^8) = C_{ijv} - \pi'_b,$$
$$f(\bar{A}_v^2), f(\bar{A}_v^5) = d_{\rho v}^g$$
$$f(\bar{A}_v^3), f(\bar{A}_v^6) = \sum_{\tilde{g}=g}^{g+\tau_{ij}} \pi'''_{\rho\tilde{g}},$$
$$f(\bar{A}_v^4), f(\bar{A}_v^7), f(\bar{A}_v^9) = C_{ijv},$$

where C_{ijv} represents the transport cost from node i to j and $d_{\rho v}^g$ represents the demurrage cost for a vessel $v \in V$ at a port $\rho \in L$ at time period g.

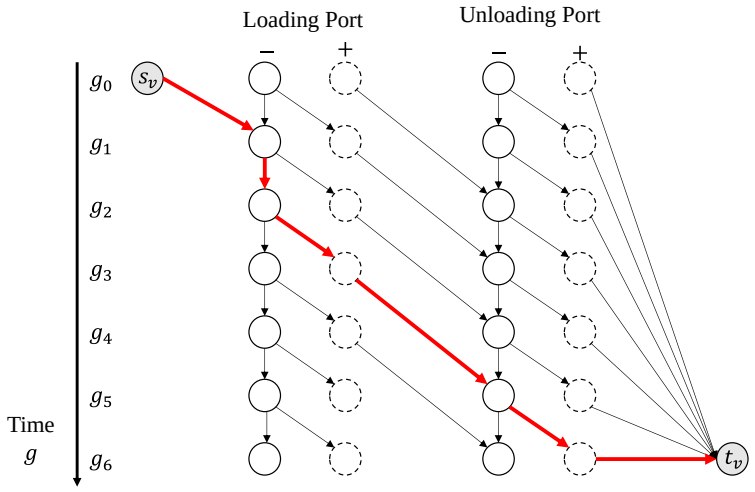


Figure 1: Time-space network with consideration of demurrage

Figure 1 shows an example of the time-space network with a consideration of berthing clash avoidance constraints and minimizing demurrage. On Figure 1, the thick red lines show a feasible s_v-t_v path for a vessel. In the path, an arc at the node for arrival loading port from time period g_1 to g_2 represents the demurrage. The key point is that we design the arcs between two arrival nodes in order to consider the demurrage. By designing arcs described above, we can naturally obtain and clearly recognize the optimal solution of the shortest path problem including the demurrage. This shortest path problem becomes a CSPP because a route on the time-space network has a possibility that the route may take the same order twice or more depending on the problems. The solution framework of the dynamic programming for this CPSS and numerical results will be reported at the conference.

References

[1] M. Christiansen, K. Fagerholt, B. Nygreen and D. Ronen, "Ship routing and scheduling in the new millennium", *European Journal of Operational Research* 228, 467–483 (2013).

[2] K-W. Pang, Z. Xu and C-L. Li, "Ship routing problem with berthing time clash avoidance constraints", *International Journal of Production Economics* 131, 752–762 (2011).

Stochastic Single-Allocation Hub Location

Nicolas Kämmerling
Institute of Transport Logistics
TU Dortmund University, Germany
Email: nicolas.kaemmerling@tu-dortmund.de

Borzou Rostami
CERC in Data-Science
Polytechnique Montréal, Canada

Christoph Buchheim
Fakultät für Mathematik
TU Dortmund University, Germany

Joe Naoum-Sawaya
Ivey Business School
Western University, Canada

Uwe Clausen
Institute of Transport Logistics
TU Dortmund University, Germany

1 Introduction

A transport network with many sources and sinks can be very expensive to operate if all shipments are transported directly from the source locations to the destination locations. To benefit from economies of scale, a number of hubs are often established to act as transshipment nodes that can handle the passing flow at a reduced cost. Hub nodes are used to sort, consolidate, and redistribute flows and their main purpose is to achieve economies of scale. While the construction and operation of hubs and the resulting detours lead to extra costs, the bundling of flows decreases the overall cost of operation. The hub location problem optimizes the location of hubs and the allocation of origin and destination nodes to the selected hubs in order to route the flow from the origin nodes to the corresponding destinations while minimizing the total cost of the network. The hub location problem arises in several important applications including telecommunication systems, airline services, postal delivery services, and public transportation, among several others.

Hub location problems are part of the strategic planning decisions and thus the exact operational data of the network is usually unknown and can only be approximated at the time the network is planned. One main source of uncertainty are the stochastic shipping volumes. As hub locations are planned well in advance of the actual operation of the network, only statistical data about shipment sizes are typically available. The usual approach of using average values makes it

difficult to give a correct estimate of the necessary hub sizes and the optimal allocation and flow routing. Thus it is often necessary to include uncertainty when deciding the location of hubs and the allocation of nodes to the hubs.

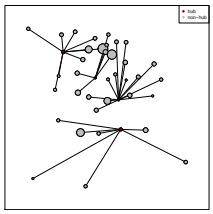
This extended abstract considers the single allocation hub location problem (SAHLP) with demand uncertainty. The single allocation problem denotes the case where each node is assigned to a single hub. We also distinguish between fixed and variable allocations. For fixed allocation, the assignments of the spokes (i.e. non-hub nodes) to the hubs are considered as part of the strategic decisions and therefore are first-stage decisions and remain fixed when uncertainty is realized. This problem variant was already introduced by Alumur et al. [1]. Alternatively, for variable allocation, the assignments of the spokes to the hubs are more flexible and can be adjusted when the uncertainty is realized and thus the allocation decisions are considered as second-stage decisions which is in line with real-world practices where the hubs are chosen before knowing the demand while the allocations are determined/altered when the actual demand is realized. Prior work has addressed the fixed allocation case, while we introduce the variable allocation stochastic hub location problem and propose a computationally efficient solution approach based on exploiting the problem formulation using cutting planes.

In Section 2, we propose a model for hub location problems with variable allocation. Further, we show how to reformulate the problem in order to solve it computationally faster. For a more detailed description of our work, please confer our preprint [2].

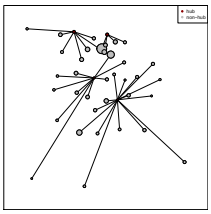
2 Variable Allocation in Hub Location

Models with fixed allocation assume that the allocation of the spokes to the hubs cannot be changed when the demand is realized. Alternatively, this section considers the variable allocation problem where the hubs are chosen before knowing the actual demand while the allocation is determined when the actual demand is realized. The advantage of taking variable allocations into account is illustrated in two examples shown in Figures 1. Each subfigure shows the choice of the hubs and the allocation from the spokes to the hubs for an example of the capacitated hub location problem. Figures 1a show the solution of the fixed allocation for a case with 5 scenarios. For each of these scenarios the individual spoke allocations are displayed in Figures 1b–1f resulting in an overall decrease of 2.0% in the objective function value. We observe in this examples that different hubs are chosen when variable allocation is used compared to fixed allocation.

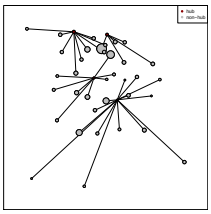
The stochastic SAHLP with variable allocation is formulated as a two-stage stochastic program with recourse. The first-stage decisions are the location of the hubs to be opened while the second-stage decisions are the optimal allocation of the spoke nodes to the hub nodes as well as the routing of the flows. To formulate the deterministic equivalent formulation of this two-stage stochastic



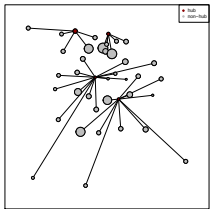
(a) Fixed Allocations for Scenario 1 to 5.



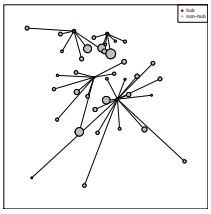
(b) Variable Allocations for Scenario 1.



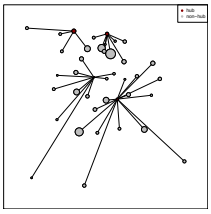
(c) Variable Allocations for Scenario 2.



(d) Variable Allocations for Scenario 3.



(e) Variable Allocations for Scenario 4.



(f) Variable Allocations for Scenario 5.

Figure 1: Fixed and Variable Allocations for a 40 node instance. The area of each node is proportional to its outgoing flow.

program with recourse, we distinguish the hub selection variables from the allocation variables. Let N be set of nodes. The selection variables are defined as the binary variables z_k, to indicate whether a hub is located at node $k \in N$ or not. The allocation variables are defined as the binary variables x_{ik}^s, to indicate whether node $i \in N$ is allocated to a hub located at node $k \in N$ under scenario s of a finite set S of scenarios occurring with probability p_s. The deterministic equivalent problem is then formulated as

$$\begin{aligned}
\text{DEF}: \quad \min \quad & \sum_{k\in N} f_k z_k + \sum_{s\in S} p_s \sum_{\substack{i,k\in N \\ i\neq k}} d_{ik} \left(\sum_{j\in N} w_{ij}^s + \sum_{i\in N} w_{ij}^s\right) x_{ik}^s + \\
& \sum_{s\in S} p_s \sum_{i,j\in N} \alpha\, w_{ij}^s \Big(d_{ij} z_i z_j + \sum_{\substack{\ell\in N \\ j\neq \ell}} d_{i\ell} z_i x_{j\ell}^s + \sum_{\substack{k\in N \\ i\neq k}} d_{kj} x_{ik}^s z_j + \sum_{\substack{k,\ell\in N \\ i\neq k, j\neq \ell}} d_{k\ell} x_{ik}^s x_{j\ell}^s \Big) \qquad (1) \\
\text{s.t.} \quad & \sum_{\substack{k\in N \\ i\neq k}} x_{ik}^s = 1 - z_i \quad i \in N, s \in S \qquad (2) \\
& x_{ik}^s \leq z_k \quad i,k \in N,\ i \neq k, s \in S \qquad (3) \\
& z_i \in \{0,1\} \quad \forall i \in N \qquad (4) \\
& x_{ik}^s \in \{0,1\} \quad \forall i,k \in N, s \in S. \qquad (5)
\end{aligned}$$

where d_{ij} are the distances between nodes i and j and w_{ij}^s is the amount of flow to be transported from node i to node j in scenario $s \in S$.

Due to the quadratic structure of DEF, a natural way to tackle this problem is to linearized it

and solving the resulting MILPs using a commercial solver. However, it is practically impossible to solve these MILPs for large-size instances in reasonable computational times.

Thus, we propose a MINLP reformulation of problem DEF as follows

$$\text{REF}: \min \quad \sum_{k\in N} f_k z_k + \sum_{s\in S} p_s \sum_{\substack{i,k\in N\\ i\neq k}} d_{ik} \Big(\sum_{j\in N} w_{ij}^s + \sum_{i\in N} w_{ij}^s\Big) x_{ik}^s + \sum_{s\in S} p_s \sum_{i,j\in N} \alpha\, w_{ij}^s \Big(u_{ii}^s z_i + \sum_{\substack{k\in N\\ k\neq i}} u_{ik}^s x_{ik}^s + v_{ij}^s z_j + \sum_{\substack{\ell\in N\\ \ell\neq j}} v_{i\ell}^s x_{j\ell}^s\Big) \tag{6}$$

$$\text{s.t.} \quad (2)-(5)$$

$$u_{ik}^s + v_{i\ell}^s \geq d_{k\ell} \quad i,k,\ell \in N,\ s\in S \tag{7}$$

$$u, v \text{ unrestricted}, \tag{8}$$

where the quadratic part of the objective function has been replaced by (6), (7), and (8).

By projecting out the variables u and v, REF can be decomposed as

$$\text{MP}: \min \quad \sum_{k\in N} f_k z_k + \sum_{s\in S} \sum_{\substack{i,k\in N\\ i\neq k}} p_s\, d_{ik} \Big(\sum_{j\in N} w_{ij}^s + \sum_{i\in N} w_{ij}^s\Big) x_{ik}^s + \sum_{s\in S}\sum_{i\in N} p_s \alpha\, \eta_i^s \tag{9}$$

$$\text{s.t.} \quad (2)-(5)$$

$$\eta_i^s \geq \psi_i^s(z,x) \quad i\in N,\ s\in S, \tag{10}$$

where for each $i\in N$, and $s\in S$

$$\text{PS}_i^\omega(\bar z,\bar x):\ \psi_i^s(\bar z,\bar x) = \max \quad \sum_{j\in N} w_{ij}^s \Big(u_{ii}^s \bar z_i + \sum_{\substack{k\in N\\ k\neq i}} u_{ik}^s \bar x_{ik}^s + v_{ij}^s \bar z_j + \sum_{\substack{\ell\in N\\ \ell\neq j}} v_{i\ell}^s \bar x_{j\ell}^s\Big) \tag{11}$$

$$\text{s.t.} \quad u_{ik}^s + v_{i\ell}^s \leq d_{k\ell} \quad k,\ell\in N \tag{12}$$

$$u, v \text{ unrestricted}. \tag{13}$$

Since $\text{MP}_{\mathbf{v}}$ is convex, and due to the fact that its objective function is linear, then the optimal solution of MP always lies on the boundary of the convex hull of the feasible set and therefore a cutting-plane approach can be used to solve the problem to optimality. More precisely, for a feasible solution $(\bar z,\bar x)$ of MP the optimal solutions of each subproblem $\text{PS}_i^\omega(\bar z,\bar x)$ provides an subgradient cut for MP. We prove that the subproblems $\text{PS}_i^\omega(\bar z,\bar x)$ can be solved efficiently. Thus, a fast cut-generating procedure is integrated in the branch-and-cut framework for solving MP.

Extensive computational results on the single allocation hub location problem and two of its variants, the capacitated case and the single allocation p-median problem are conducted on AP instances with up to 200 nodes. The proposed cutting plane approach outperforms the direct solution of the problem using the state-of-the-art solver GUROBI. A detailed discussion on the computational results can be found in our preprint [2].

References

[1] S.A. Alumur, S. Nickel, F. and Saldanha-da-Gama, "Hub location under uncertainty", *Transportation Research Part B: Methodological* 46, 529-543 (2012).

[2] B. Rostami, N. Kämmerling, C. Buchheim, J. Naoum-Sawaya, and U. Clausen, "Stochastic Single-Allocation Hub Location", Technical report, CERC, 2018.

Tactical Design of Same-Day Delivery Systems

Alexander M. Stroh, Alan L. Erera, Alejandro Toriello*

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, Georgia, USA

October 15, 2018

Introduction and Problem Formulation

E-retail continues to grow apace in the US and around the world. According to a recent report from eMarketer, e-retail makes up roughly 10% of all US retail sales, with Amazon alone accounting for almost half of this number. Within this sector, competition is fierce and companies continuously search for ways to deliver goods to customers faster and more economically. A popular emerging service is *same-day delivery* (SDD), where customers receive their orders the same day the orders are placed; in a 2017 survey of 500 North American retailers by BRP consulting, 51% claimed to offer some form of SDD, up from 16% reported in 2016.

Over the past few years, the transportation and logistics research community has proposed and studied operational policies for SDD distribution systems under a variety of models and assumptions, e.g. [3, 4, 5, 7]. The models considered therein typically assume a fixed SDD system design, including service area, delivery vehicle fleet size, time the service is offered, etc., and perform a detailed analysis, optimization and/or simulation of operating policies. In contrast, the research community has to our knowledge not focused on the SDD distribution system's tactical design variables: How large should the SDD vehicle fleet be? How late in the day should SDD be offered to customers? How big should the service area be? We are not aware of any research tackling these and other important tactical-level questions; this paper's goal is to begin addressing this gap.

Our model captures the "average" behavior of a single SDD dispatch facility over an operating day, where orders are packed for final delivery and dispatched on delivery vehicles. As in most e-retail settings, we assume orders are customer-specific and can only be dispatched after they appear; we also assume a common delivery deadline, e.g. the end of the business day, rather than the order-specific deadlines more common in restaurant delivery. As SDD has tight delivery deadlines and comparatively low order volume, time (not vehicle capacity) is the constraining

*corresponding author, atoriello@isye.gatech.edu

resource, so we focus more on timing and route duration and less on package size, weight and capacity.

The model has the following components. A single depot with m vehicles provides SDD service to a service region with area A. SDD orders appear at a constant rate of λ per unit of area and time in random locations throughout the service region, starting at time 0 and ending at time N, the latest time an SDD order can be placed. The deliveries can start any time after 0 and must finish (with all orders delivered and vehicles back at the depot) by an operating deadline $T > N$. Without loss of generality, we take $A = \lambda = 1$, so that N is the total number of SDD orders to deliver. We also assume here for simplicity of exposition that the service region is not subdivided and dispatches serve the entire region. Our model can have many reasonable objectives; we mostly focus here on minimizing the total vehicle dispatching time.

We model dispatch times with a generic continuous approximation, where delivering $n \in \mathbb{R}_+$ orders to the service region takes $f(n)$ units of time and f is concave, increasing, and satisfies simple technical conditions. Our motivation is functions of the form $f(n) = an + b\sqrt{n}$, where the linear term models service time per order and the root term is a BHH approximation [1]. Continuous approximations are used widely in logistics [2] and have recently been successfully applied in a last-mile operational context [6].

Optimal System Behavior

We focus our analysis on two important fleet size cases: when the fleet is large enough to require at most one dispatch per vehicle (m large); and when a single vehicle makes all deliveries ($m = 1$).

In the large fleet case, the following dispatch policy is optimal: Send the first dispatch at time t satisfying $t + f(t) = T$, i.e. when it has exactly enough time to take all accumulated orders and return at T. For the second dispatch, repeat the process with all orders accumulating after the first dispatch; continue until a dispatch can take all remaining orders and depart after time N. The optimal dispatch times for each vehicle can be computed by solving equations of the form $t + f(t) = T'$ for some $T' \leq T$.

In the single-vehicle case, under an additional technical assumption, there is an optimal policy with the following structural properties: (1) each dispatch takes all available orders at the depot at the time of dispatch, and (2) after the first dispatch, the vehicle never waits at the depot again, finishing precisely at time T. This structure implies that the dispatch times can be computed via an optimization model over a single variable, t, defined as the time of first dispatch. One can solve for the optimal t via an iterative root-finding algorithm, and then the second dispatch time is $t + f(t)$, the third is $t + f(t) + f(f(t))$, and so on.

The proofs in both cases use the concave and increasing structure of f, which intuitively imply that we minimize routing time by dividing orders into dispatches as unevenly as possible, i.e. so that

the first dispatch takes as many orders as possible, the second as many as possible of the remainder, and so forth. The additional assumption for the single-vehicle case states that all dispatches except possibly the last one must deliver a sufficiently large number of orders (that is, early dispatches do not deliver small order numbers), and that there must be a sufficient gap between the order deadline N and the service deadline T; both assumptions are justifiable as common sense business rules.

Example A retailer provides SDD service over an 8 mile by 8 mile service region, with an average of 75 orders arriving from uniformly random delivery locations over a 10-hour SDD service window. Retailer operations begin with the service window and last 12 hours; this translates to $N = 75$, $T = 90$. Take the routing time function $f(n) = 0.13n + 2.15\sqrt{n}$, which is roughly equivalent to a BHH approximation with rectilinear (Manhattan) distances and average vehicle speed of 25 mph, plus a service time of 1 minute per order.

If the fleet has multiple vehicles, only two vehicles are required in this setting. The first dispatch occurs after approximately 64 orders accumulate, with the second covering the remaining 11, with total dispatching time of 272 minutes. If one vehicle must make all the deliveries, the first dispatch would carry 55 orders, and the second the remaining 20, with a total dispatching time of 283 minutes. A manager evaluating this system should expect a daily dispatch time increase of only 11 minutes (about 4%) if they decrease the SDD fleet size from two to one vehicle.

Conversely, a manager evaluating the two-vehicle setting could instead notice that the second dispatch has a slack of 52 minutes. This could motivate an increase in the service window length to 79 orders, or roughly 10.53 hours, which could still be served by two vehicles over the same 12-hour operating day, with total dispatch time of 286 minutes.

The example motivates a profit maximization version of our model, which we have also analyzed. Instead of holding N fixed, we include it in the objective with a linear term representing revenue gained by SDD orders served, and optimize the profit given by this revenue minus the SDD cost, represented by the system's total dispatch time. In the large fleet case, the general result is analogous to our example: As a function of N, the profit is piecewise convex, with breakpoints where the number of vehicles required to serve all orders increases; one such breakpoint is always the optimal choice. In the single-vehicle case, the corresponding breakpoints are the order numbers at which the total number of dispatches required to serve the orders increases; our analysis so far has verified similar results when the number of dispatches is one or two, but we expect the result to hold in general, as in the many vehicle case.

In conclusion, we propose a tactical-level model for SDD systems. Our model predicts the average behavior of the system under various settings, and allows managers to evaluate the impact of system design variables such as fleet size and service window length. Our ongoing work includes

validating our analytical results by testing them in an operational context. Using a model similar to [3], we can construct operational instances that inherit the parameters of our tactical model. We will then simulate operations over many days, to test the predictions our tactical model makes with respect to dispatch time, number of dispatches, etc.

References

[1] J. Beardwood, J.H. Halton, and J.M. Hammersley, *The shortest path through many points*, Mathematical Proceedings of the Cambridge Philosophical Society **55** (1959), 299–327.

[2] A. Franceschetti, O. Jabali, and G. Laporte, *Continuous approximation models in freight distribution management*, TOP **25** (2017), 413–433.

[3] M. Klapp, A.L. Erera, and A. Toriello, *The Dynamic Dispatch Waves Problem for Same-Day Delivery*, European Journal of Operational Research **271** (2018), 519–534.

[4] ———, *The One-Dimensional Dynamic Dispatch Waves Problem*, Transportation Science **52** (2018), 402–415.

[5] M.W. Ulmer, B.W. Thomas, and D.C. Mattfeld, *Preemptive Depot Returns for Dynamic Same-Day Delivery*, EURO Journal on Transportation and Logistics (2018), Forthcoming.

[6] W.J.A. van Heeswijk, M.R.K. Mes, and J.M.J. Schutten, *The delivery dispatching problem with time windows for urban consolidation centers*, Transportation Science (2017), Forthcoming.

[7] S. Voccia, A.M. Campbell, and B.W. Thomas, *The same-day delivery problem for online purchases*, Transportation Science (2017), Forthcoming.

An Intermodal Hub Location Problem for Container Distribution in Indonesia

Hamid Mokhtar

School of IT and Electrical Engineering, The University of Queensland, Australia
Email: h.mokhtar@uq.edu.au

A. A. N. Perwira Redi

Department of Mech. & Aero. Eng., Monash University, Clayton VIC 3800

Mohan Krishnamoorthy

School of IT and Electrical Engineering, The University of Queensland, Australia

Andreas T. Ernst

School of Maths. Sciences, Monash University, Clayton VIC 3800

1 Introduction

Hubs, arising in several network design contexts which involve commodity flow interchange between nodes and transfer modes, are centralised facilities for aggregation and dis-aggregation of flow between nodes, and/or handling goods for change of transfer mode. In such network design problems, the transfer modes may represent different transport vehicle types that carry goods between nodes and hubs and between hubs. A hub location problem determines a set of hubs and designs a set of routes through hubs to fulfil the origin-destination flow demands at least cost.

In many cases, these problems arise when, in contrast to common assumptions, hubs are not fully connected, an access node is not necessarily directly connected to a hub, or any route can interchange flow between multiple transfer modes. Here, we present a dataset for the intermodal hub location problem, drawn from a real-world case study, which involves a sparse network, three different transport modes, and two types of hubs. We introduce and study an intermodal hub

location problem, with the archipelago of Indonesia as the context. Our study provides a strategic solution for flow through the integrated use of two or more modes of transportations.

2 The ICD Dataset

We develop the Indonesia Container Distribution (ICD) dataset, based on real data from the Indonesian container transportation network. The potential hubs include five inland terminals, which are hubs that enable transfers between road and rail modes of transport. Compared to the benchmark datasets in the hub location problem literature, the ICD dataset offers various and different features which can be used while testing new models and algorithms for HLP. The main differences are the consideration of intermodal transportation and the sparsity of network in the instances. Moreover, the sparsity of network in the ICD dataset reflects the presence of a complex transportation network in real world problems.

3 The Intermodal Hub Location Problem (IHLP)

We are given a direct network (N, A), where A is the set of arcs and $N = \{1, 2, \ldots, n\}$ is the set of nodes. Arcs can be partitioned into subsets A_1, A_2, and A_3, respectively for arcs with sea, road, and rail transfer modes. We are also given flow demands W_{ij} from i to j, and the length d_e of arc $e \in A$. In this network, any two adjacent arcs in a path must be either in the same mode, or be incident to a hub node. A hub is a node which facilitates a transfer mode interchange at some cost. A hub can be selected from a given subset $H \subseteq N$ of nodes. The problem of locating a set of hubs among n nodes, and routing each flow demand with minimum total cost is called the *intermodal hub location problem* (IHLP). We may include potential inland terminals and rail links in our dataset to study the impact of network expansions. In the IHLP, the number of located hubs, including seaports and inland terminals is to be decided in such a way that the total cost is minimised. For a given pair of integers (q, p), where $q \leq p$, the intermodal multiple allocation hub location problem is one in which the *total* number of hubs is fixed to p and the minimum number of inland terminals is set to q, denoted as the (q, p)-*intermodal hub median problem* ((q, p)-IHMP).

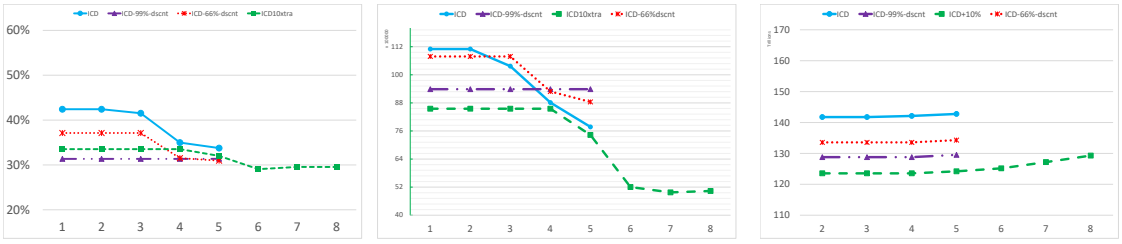
Theorem 3.1. *(q, p)-IHMP is NP-hard.*

4 Analysis of Network Design

In on our extensive computational results on the ICD dataset [1], we consider and analyse different scenarios (for instances with 66 and 73 nodes). The scenarios we considered are: (i) the original dataset (ICD), (ii,iii) `ICD-66%-discnt` and `ICD-99%-discnt`, respectively, the pricing of train

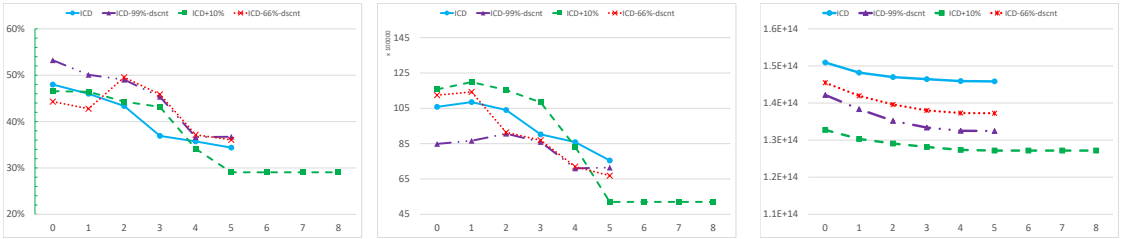
transportation is discounted by 66% and 99%, and (iv,v) networks with more rail links incident to the 10%, and 25% highest demand nodes (`ICD+10%` and `ICD+25%`). Our experiments with the various scenarios provide us insights into pricing options on different transfer modes and investments on network infrastructure. For instance, it is inferred from our experiments that seaports are less congested when either more inland terminals (and associated rail links) are installed, or when a discount factor is applied for rail transportation. It is what we would expect, intuitively. However, the model and the results demonstrate that this is indeed the case.

As shown in Figure 1, when p is fixed, the traffic congestion on seaport roads decreases as q grows in all scenarios for $q \leq 5$. When train costs are well-discounted or new rail links are added to the network, simultaneously the number of containers shipped through seaports declines and the usage of trains becomes more attractive. So the flow on road links in the network is deviated towards inland terminals. In scenarios `ICD-66%-discnt` and `ICD-99%-discnt`, the road congestion on seaport roads decrease as q grows. However, once q becomes too large so that the number of seaports drops to 5, the usage of trains is not cost effective, even if more rail links for high demand nodes are added in the network.



(a) Road usage percentage (b) Road Load (c) Total costs

Figure 1: Road and rail usage at seaports as q increases for $n = 66$ and $p = 15$

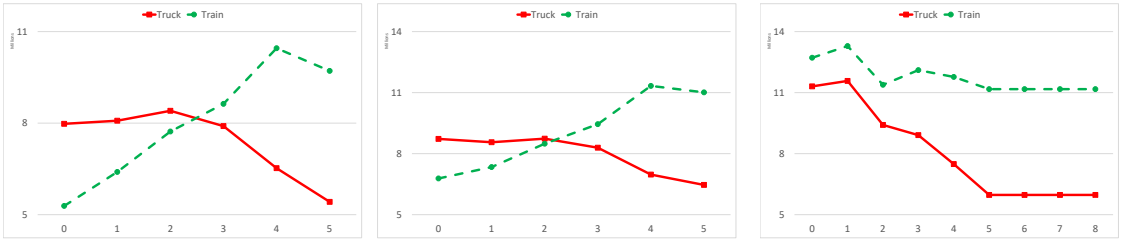


(a) Road usage percentage (b) Road load (c) Total costs

Figure 2: Road and rail usage at seaports as q increases for $n = 66$ and $p - q = 9$

Now we analyse the impact of adding an inland terminal to the set of hubs. By Figure 2, the

proportion of road congestion at seaport links is considerably decreased and the total operational and fixed costs decreases by at most 5%. There is an increase in road transfer to seaports in most scenarios when the number of inland terminals reaches to 2 because there is a better opportunity to substitute road transfer by rail transfer in a few islands, but the proportion of road usage remains declining as q grows. Besides installing more inland terminals, the facilities of rail usage (by new rail links, or discounted rails) promotes the ratio of the rail usage to the road usage. By Figure 3, there is a shift in the number of containers shipped to seaports by road to rail as q grows.



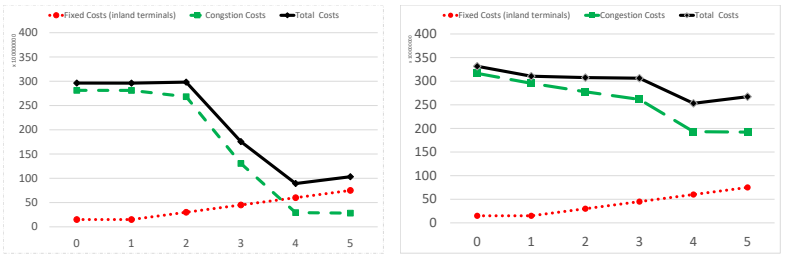
(a) ICD (b) `ICD-66%-dscnt` (c) ICD+25%

Figure 3: Road and rail usage at seaports as q increases for $n = 66$ and $p - q = 8$

In Figure 4, we compare the fixed costs and congestion costs (using an economic model for congested roads) in seaports. The congestion cost is associated with the opportunity cost of delays on roads. It is proportional to the number of road users and reciprocal of the average speed on roads when the capacity of roads is fixed.

References

[1] Hamid Mokhtar, A.A.N. Perwira Redi, Mohan Krishnamoorthy, and Andreas T. Ernst. An intermodal hub location problem for container distribution in indonesia. *Computers & Operations Research*, 2018.



(a) $n = 66$ (b) $n = 73$

Figure 4: Trend of congestion costs and fixed costs for new inland hubs