TRISTAN II

TRIennal Symposium on Transportation ANalysis

PREPRINTS

C.N.R
Progetto Finalizzato
Trasporti Due

C.N.R.-I.A.S.I.
Istituto di Analisi
dei Sistemi ed Informatica

A.I.R.O. Associazione Italiana
Ricerca Operativa

Vol. I

CAPRI, ITALY
Congress Center
June, 23-28 1994

TRISTAN II

TRIennal Symposium on Transportation ANalysis

PREPRINTS

Vol. I

This compendium was prepared in the interest of timely information dissemination. It contains the preprints of the papers presented by the authors during the Conference.

Capri, June 23-28, 1994

TRISTAN II

TRIennal Symposium on
Transportation ANalysis

PROGRAM COMMITTEE

Lucio Bianco	*Chairman*	University of Rome Tor Vergata, Italy
Paolo Toth	*Chairman*	University of Bologna, Italy
Maurizio Bielli		I.A.S.I. - C.N.R., Italy
Ennio Cascetta		University of Naples, Italy
Teodor Gabriel Crainic		University of Montreal, Canada
Warren B. Powell		Princeton University, USA

ADVISORY COMMITTEE

Michael O. Ball
Jaime Barcelo
Alex Belenky
Carlos Daganzo
Mark S. Daskin
Pierre Dejax
Jacques Desrosiers
Awi Federgruen
J. Enrique Fernandez
Michael Florian
Terry L. Fries
Giorgio Gallo
Patrick T. Harker
Toshihide Ibaraki
Martine Labbe'
Gilbert Laporte
Jan Karel Lenstra
Amedeo R. Odoni
Jose' Paixao
Markos Papageorgiou
Harilaos N. Psaraftis
Donald H. Ratliff
Alexander Rinnooy Kan
Werner Rothengatter
Jean-Marc Rousseau
Francois Soumis

ORGANIZING COMMITTEE

M. Bielli   G. E. Cantarella   A. Scognamiglio

INVITED LECTURERS

J. Barcelo	University of Barcelona, Spain
M. Ben-Akiva	Massachussets Institute of Technology, Cambridge, USA
E. Cascetta	University of Naples, Italy
M. Florian	Centre for Research on Transportation, Montreal, Canada
W. Powell	Princeton University, USA
H.D. Ratliff	Georgia Tech., Atlanta, USA

SECRETARIAT of TRISTAN II

- **Maurizio Bielli** I.A.S.I. - C.N.R.
  Viale Manzoni, 30 - 00185 - Rome - Italy
  Phone + 39 - 6 - 77161 - Fax + 39 - 6 - 7716461
  E-MAIL: BIANCO@IASI.RM.CNR.IT

Volume stampato con il contributo del Consiglio Nazionale delle Ricerche, Comitato Nazionale Scienze di Ingegneria ed Architettura

INDEX

VOLUME I

TUTORIALS

PLENARY SESSIONS

PARALLEL SESSIONS

Heuristics for Vehicle Routing Problems (*Chairman:* J.M.Rousseau)

Planning Models for Urban Traffic Systems (*Chairman:* M. Papageorgiou)

Production, Transportation and Logistics (*Chairman:* H.Psaraftis)

Probabilistic Vehicle Routing Problems (*Chairman:* A. S. Belenky)

IVHS Applications (*Chairman:* P. Mirchandani)

TRANSPORT ECONOMICS (*CHAIRMAN*: T. IBARAKI)

RESOURCE MANAGEMENT (*CHAIRMAN*: G. LAPORTE)

DYNAMIC ASSIGNMENT (*CHAIRMAN*: B. SANSO')

TRANSPORT NETWORK ANALYSIS (*CHAIRMAN*: A. FEDERGRUEN)

DESIGN AND EVALUATION OF PUBLIC TRANSPORT ROUTES (*CHAIRMAN*: J.E. FERNANDEZ)

NONLINEAR NETWORKS (*CHAIRMAN*: P. JAILLET)

VOLUME II

Modelling Dynamics in Traffic Assignment (*Chairman*: M. Labbe')

Crew Scheduling Problems (*Chairman*: J. Desrosiers)

Algorithms for IVHS (*Chairman*: J. Barcelo)

Urban Transportation Planning (*Chairman*: C. Barnhart)

Methods for Vehicle Routing Problems (*Chairman*:A. Wren)

Air Transportation (*Chairman*: M. Bielli)

Equilibrium and Assignment Models (*Chairman*: Y. Asakura)

Relaxation Methods for Vehicle Routing Problems (*Chairman*: F. Soumis)

Signal Control (*Chairman*: G. E. Cantarella)

Network Design (*Chairman*:H.S. Mahmassani)

Special Cases of Vehicle Routing Problem (*Chairman*: A. *Mingozzi*)

Traffic Planning Models (*Chairman*: S. Pallottino)

Public Transport (*Chairman*: D. H. Ratliff)

Algorithms for Vehicle Routing Problems (*Chairman*: N. Christofides)

Transport Planning Models (*Chairman*: J. Bramel)

TUTORIALS

Dynamic Models in Transportation

Warren B. Powell
Princeton University

The field of logistics is becoming increasingly dominated by the need for technologies that support real-time decision making. Using recent advances in information technologies, decisions concerning the routing and scheduling of drivers and vehicles, management of vehicle inventories, and the design of service offerings, can be made in a real-time environment with information that is constantly changing. Dynamic and stochastic models are playing an increasingly important role in such a setting: by definition, one is forced to make decisions before all the information one would wish to have becomes available and then modify these decisions as new information is received. The most common form of stochasticity arises as a result of uncertainty concerning some aspect of demand (level of demand, location, timing, etc.) but many other forms may be present, as well (length of travel times, resource availability, service breakdowns, etc.). As a rule, once a set of decisions has been made and some action has been taken, the decision-makers have the opportunity to observe an outcome of (some of) the uncertain events and must then respond to these events. The inherent dynamism of this type of operation often introduces important analytical complications: initial decisions may be greatly affected by how well the decision- makers –and the system they manage– are equipped to respond to subsequent random events.

Dynamic models undoubtedly represent the "wave of the future." Their increasing prominence is driven by technology: the explosive growth in the availability of real- time information about transportation and logistics systems is turning the focus of operations researchers away from the traditional static planning models. Carriers and shippers are avidly seeking the benefits afforded by the ability to rapidly and continually "reconfigure" the operation of a transportation system to improve service or reduce cost. In addition, even when it comes to strategic planning, methodological developments over the last few years have made possible the development of models that capture better the uncertainty and the dynamic characteristics associated with the transportation and logistics environment. Clearly such models often constitute far more accurate representations of reality than some of their deterministic and static forebears.

Especially demanding are multistage problems where the process of making decisions and observing outcomes occurs on a continuous, rolling basis. A set of vehicles may be routed over the course of the day while demands are continuously being called in. With each new demand, the vehicle tours may be redesigned, not only to accommodate actual demands, but anticipated demands as well. Another example arises in dynamic fleet management, where empty vehicles are repositioned to anticipate future demands. Such decisions are made daily, as shipper demands are continuously being received at a dispatching center.

One question that comes up with surprising frequency is: what constitutes a dynamic model? To answer this, we must first distinguish between a problem, a model, and the application of a model. A problem is dynamic if one or more of its parameters is a function of time. This includes such problems as vehicle routing with time windows or with time-varying travel times. Note that two types of dynamic problems are covered here. The first type, which we call problems with dynamic data, are characterized by information that are constantly changing. Dynamic data might include real-time customer demands, traffic conditions, or driver statuses. The second type is problems with time-dependent data which is known in advance. In this category, we would include problems such as vehicle routing with time windows where all the information is known in advance, but where this data is a known function of time. Other examples of time-dependent data might be customer demands or travel times, if we can assume them to be known functions of time.

Similarly, a model is dynamic if it incorporates explicitly the interaction of activities over time. The simplest dynamic model is a dynamic network, a construct widely used in routing and scheduling problems. It is useful, however, to distinguish between deterministic, dynamic models, and stochastic models which explicitly capture the *staging* of decisions and the realization of random variables. For example, it is not unusual to solve deterministic, dynamic models without recognizing in any way the dynamic structure of the problem. By contrast, stochastic, dynamic models require specific steps to be taken in the design of a solution strategy.

Finally, we have a dynamic application if a model is solved repeatedly as new information is received. Dynamic applications of models place tremendous demands on access to real-time data and on the performance of algorithms. Typically, it is necessary to update information, optimize and return results in a matter of minutes or even seconds.

A useful illustration of these concepts arises in the problem of routing a vehicle over a congested transportation network. Clearly, traffic conditions are changing over time, and hence the problem is dynamic. We might select an optimal route using a static model, if we chose not to represent dynamic conditions explicitly within the model. (For example, we could find the optimal route by simply minimizing average travel times, thus working with one particular static representation of the network.) We might also solve the static model repeatedly as new information became known, giving us a dynamic application of a static model. Yet a third alternative might be to develop a model that would consider explicitly the anticipated dynamic changes in traffic conditions and apply it once at the outset of the period of interest. In other words we would select one route and then stick to it even as conditions change. This would constitute a static application of a dynamic model. Finally, had we chosen to solve the dynamic model repeatedly as new information became known, we would have a dynamic application of a dynamic model. This example, then, indicates how one can have a static or dynamic application of a static or dynamic model.

In this tutorial, we review important dynamic planning problems, and review modeling and algorithmic strategies that can be used. Special attention is given to myopic models, deterministic dynamic models, and explicit stochastic models. Emphasis is placed on computationally tractable techniques that have practical applications. Sources of uncertainty are discussed, and special attention is given to the pros and cons of ignoring future information.

Dynamic models pose a number of interesting challenges, ranging from model formulation to validation and implementation. We contrast dynamic models with static models, and discuss the ways that "static thinking" pervades our approach to problems. While the field of operations research has developed powerful tools, using them in a dynamic setting often involves a difficult transition to a different set of priorities. Drawing from a number of actual projects involving the implementation of dynamic routing (some successful, some not) we review challenges that arise practicing our profession in a real-time setting.

RECENT ADVANCES IN METHODS AND MODELS IN URBAN TRANSPORTATION PLANNING

Michael Florian, C.R.T. and Department d'INRO, University of Montreal, Canada

Abstract: The models and methods used in the quantitative analyses employed for urban transportation planning studies have been the subject of considerable refinement over the past few years. From the large body of academic research in the area of network equilibrium models a considerable transfer to practice has occurred. While the gap between the theoretical research results available and the methods that have found their way to practical application is still large, the level of sophistication of models used in practice has increased. The purpose of this paper is to identify, describe and illustrate with selected applications some of the more advanced methods that have been successfully used.

One type of network equilibrium model that has found a variety of applications is the multiclass network equilibrium model which distinguishes different classes of users. While the model was first proposed in the scientific literature in the early '70s it is only during the past few years that several applications became widespread. In the U.S.A. the model was used for analyzing traffic on high occupancy vehicle (HOV) which has access to privileged lanes on urban auto routes simultaneously with traffic of low occupancy vehicles (LOV) which do not have access to the HOV lanes. In Europe and Central America the model found an important application in the forecasting of traffic on toll highways in and around urban areas. Stated preference analyses are used to determine the value of time for various socio-economic groups. This value of time is then used in multiclass network equilibrium models with generalized cost to predict the expected use of these new facilities. The mathematical formulation of such models is presented in detail.

Bi-level optimization problems are among the most difficult problems in the field of mathematical programming. One such problem is the adjustment of an origin-destination matrix from observed flows when the network flows are equilibrium flows. In spite of the inherent theoretical difficulty of such problems and approximate solution algorithm, which may be classified as a Gauus-Seidel like decomposition, has proved to be very effective in solving very large scale problems. The importance of being able to solve such problems stems from the fact that home interview surveys are rather costly and the update of an existing origin-destination matrix by using the most current data on observed counts is effective and cost efficient. The mathematical formulation and properties of this bi-level optimization problem is discussed in detail and results, originating from cities in Europe and North America, are used for illustrating the quality of the results obtained.

The classical way of specifying demand models for urban transportation planning models has undergone many improvements and fundamental changes. Rather than rely on the sequence of trip generation, trip distribution and mode choice models calibrated on aggregate zone data, which are not particularly sensitive to policy decision variables, a variety of demand models, which are calibrated by using disagregate data, have come into wide use. Another aspect of demand modelling which is now considered to be important is the phenomenon of trip chaining, where a trip from an initial origin to a final destination contains intermediate destinations. Perhaps the simplest model of trip chaining is a network equilibrium model of travel on combined modes such as "park and ride". This model may be specified with a hierarchical mode choice function. A proper network equilibrium model may be stated and an equivalent optimization problem, solvable by a convergent algorithm, may be derived. We present in detail such combined mode equilibrium problems and show how the analysis may be extended to more general trip chaining models.

The presentation is concluded with an outline of other research results in the area of network equilibrium models which may eventually find successful application in the process of urban transportation planning.

MODELING DYNAMICS IN TRANSPORTATION NETWORKS

by Ennio Cascetta
Dept. of Transportation Engineering
University of Naples Federico II

1. Introduction

Traffic assignment models are used to simulate link flows on transportation networks and the resulting link performances, such as travel times, congestion, pollution, and energy consumption. They are the basic tool for long term and short term planning and design of both urban and extra-urban transportation networks. Recently, on-line applications have been also proposed for supporting real-time control operations.

Most traffic assignment models share a common structure made up by:

- a *supply model* simulating the network performances;
- a *demand model* simulating users' behaviour
- a *supply/demand interaction model* simulating the interaction between users' behaviour and network performances.

Two types of dynamics (i.e. variations over time) are relevant for the analysis of transportation networks:

* evolution from one reference period to the other, called day-to-day dynamics, which mainly affects supply/demand interaction,
* variations within a reference period, called within-day dynamics, which mainly affect supply and demand models.

1.1 Day-to-day dynamics

Traditionally traffic assignment models have been formulated following an *equilibrium* approach in which a self-reproducing or fixed-point state of the system is searched. This approach relies on elegant and well developed mathematical foundations which can be effectively solved for large-scale networks, at least for the within-day static case. Furthermore, the equilibrium approach does not require the explicit modeling of users' memory and learning processes, since only one state of the system is looked for, independently from the sequence of states needed to reach it.

However equilibrium analysis is significant under some assumptions on its "significance" (coincidence or closeness with the actual state of the system) and analytical properties, such as existence, uniqueness and stability. Moreover, transients due to modifications of demand and/or supply cannot be simulated through equilibrium models, nor a statistical description of the system state, i.e. means, modes, moments and, more generally, frequency distributions over time can be obtained. This implies that dynamic control strategies (such as adaptive traffic lights, variable message signs, route or parking guidance systems) reacting to perturbations in demand and/or supply can not be effectively simulated through an equilibrium approach.

The *inter-periodic (or day-to-day) dynamic* approach can be seen as a generalization of the equilibrium paradigm. First of all, it allows the simulation of some relevant aspects such as transients, temporal fluctuations and multidimensional dynamics with different "propensity to change" over different choice dimensions (e.g. activity location, trip frequencies and distribution, mode and path choices). Moreover inter-periodic dynamic models can be seen as a tool for the analysis of theoretical properties of system convergence to different attractors (not necessarily equilibrium or fixed-point) such as existence, uniqueness and stability. From this view-point these models could be also called *disequilibrium* models, whilst the equilibrium models could be considered *day-to-day (or inter-periodic) static.*

This wider generality is obtained at the expenses of an extra complexity since an explicit modeling of the system adjustment mechanism is required, including users' memory and learning processes and their interaction with operating control strategies. On the other hand, only in this way the role of habit and non-compensatory behavior in users' choice, needed to effectively assess dynamic control strategies, can be explicitly simulated.

Two types of day-to-day dynamic process models can be formulated. *Deterministic process* models, based on non-linear dynamic system theory, can be used to analyze the asymptotic behavior of the system. They can be also used to study equilibrium properties, since the equilibrium state can be seen as a fixed-point attractor of a deterministic process under some hypotheses on users' learning mechanisms and switching behavior. *Stochastic process* models based on stochastic process theory, allow an explicit simulation of the intrinsic randomness of both demand and supply.

1.2 Within-day dynamics

If travel demand is assumed (approximately) constant over a reference period (e.g. morning peak period) which is large enough to allow the system to reach a stationary flow pattern, the assignment model is called *within-day or infra-periodic static.* Time-dependent demand (due for instance to rush hour) and/or changes in supply (due for instance to incidents or weather conditions) generally determine time-dependent flows and over saturation queues, which can be only simulated through *within-day or infra-periodic dynamic* models. These models also allow to take into account the effects of real-time control strategies (such as variable message signs, radio broadcasting, etc.).

The extension of within-day static models to take into account within-day dynamics is by no means straightforward, since within-day dynamic supply modelling requires entirely new definitions of relevant variables and a reformation of the problem, even though within-day dynamic users' behaviour can be modelled through an extension of the within-day static case.

1.3 *Classification of dynamic assignment models*

Since the types of dynamics affect different sub-models several assignment models can be classified according to the scheme shown in the following table.

	day-to-day static	*day-to-day dynamic*	
	equilibrium	deterministic process	stochastic process
within-day static			
within-day dynamic			

2. CONTENTS OF THE TUTORIAL

In this tutorial inter-periodic or day-to-day dynamics is addressed first. In particular at first a general formulation, including most within-day static models in literature, is proposed. Supply and users' behavior models, including both learning, forecasting and choice processes are described. Then several general and simplified formulations of deterministic process models and their fixed-point attractors, giving also conditions for their coincidence with equilibrium models are presented.

Within the framework of deterministic processes the relevance of day-to-day dynamic models for demand/supply interaction in comparison with the traditional user equilibrium approach is discussed, and conditions for coincidence of fixed-point attractors and equilibrium states are stated.

Conditions for existence and uniqueness of fixed-point attractors are proposed, generalizing and extending those presented in literature for user equilibrium. Conditions for stability of both fixed-points and equilibrium states are formulated by making use of results from non-linear dynamic system theory. Moreover, it is possible to devise a new family of "dynamic" algorithms which simulate the system convergence to a fixed-point in order to obtain an equivalent equilibrium state, as opposed to conventional "optimization" algorithms. In this case the fixed-point stability analysis can be viewed as a convergence analysis for the so specified algorithms.

Conditions for stochastic process regularity ensuring, among other things, existence and uniqueness of a stationary probability distribution of system states are proposed. These conditions generalize and extend results presented in literature to a wider class of possible dynamic models. Relationship between a deterministic process, together with corresponding fixed-points or equilibrium states, and stochastic probability distribution are also briefly addressed.

Then, both deterministic and stochastic process models for within-day dynamic traffic assignment are discussed, and a consistent formulations of within-day dynamic equilibrium is described. Demand models including departure time choice as well as path choice are briefly discussed.

As already noted the main difference between within-day static and dynamic models is relative to the supply side requiring the explicit and consistent definitions of link and path flows and costs and relationships among them. In particular the network loading (NL), that is the relationship between path and link flows, is linear and defined by the network topology only for within-day static networks, but highly non-linear for within-day dynamic networks. Several approaches have been proposed in literature for the solution of the dynamic network loading (DNL), and some of them will be briefly discussed and compared in the tutorial.

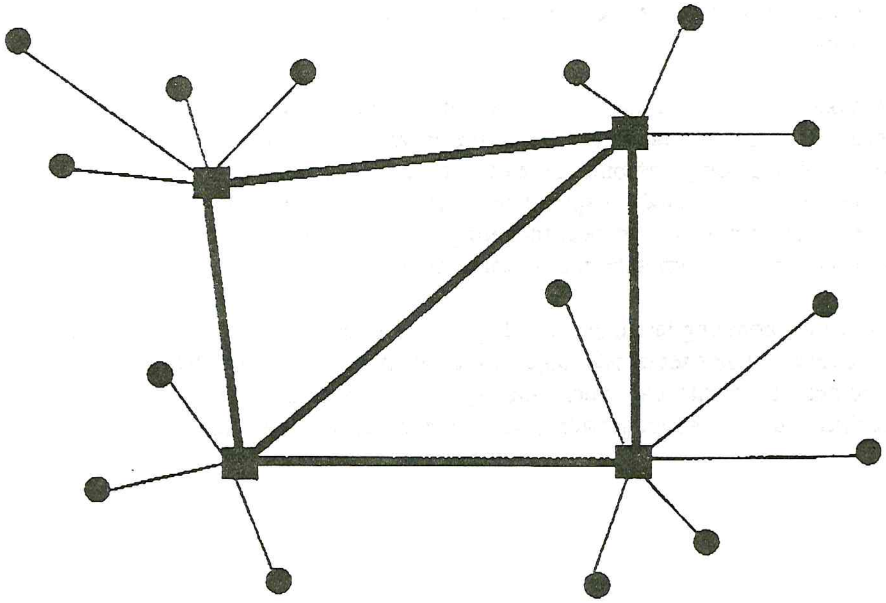
Intermodal Network Design

H. Donald Ratliff
UPS Professor - Georgia Tech
Logistics Engineering Center Director - Georgia Tech
Chairman & CEO - CAPS LOGISTICS

Introduction

Intermodal movements have the property that they use at least two different transportation vehicles in moving from origin to destination. Often the vehicles are fundamentally different. For example, the shipment of containers from Japan to Atlanta utilizes a combination of truck, train and ship. However, we also consider movements which use two vehicles of same type as intermodal movements. An example is a supplier of parts in Atlanta shipping truck loads of parts to "pooling" points in California from which they are delivered on different trucks in less-than-truck loads to the customers. The two examples above are representative of the complexity "extremes" of intermodal networks with pooling operations representing the simplest and ocean shipping the most complex.

A defining characteristic of intermodal networks is that they contain a "core" network with one or more "access" networks attached to it as illustrated in the figure below.



Intermodal network design problems are concerned with the location and configuration of terminals where transfers between vehicles occur and the location and configuration of channels connecting these terminals. The location of terminals and channels are combinatorial optimization problems with differing mathematical structures depending on the context. Configuration of terminals varies from the design of complex sorting, storage and retrieval operations at container seaports to simple truck loading at pooling points. Configuration of channels varies from the design of schedules and routes for intermodal trains and container ships in ocean shipping to the specification of number of trucks required for the truck load part of a pooling operation.

Decision technology

In order for decision technology to have a really significant impact on intermodal network design it must combine optimization concepts and algorithms, integrated data analysis and graphical user interfaces in easy-to-use tailored systems. These systems should increase the user's understanding of the intermodal system as well as aid in making specific decisions.

There are three software alternatives for decision technology: (1) custom development, (2) off-the-shelf packages, and (3) modeling languages. The biggest problems with custom development for intermodal network design are the expense to build and the long development lead time. The diversity of issues makes it difficult to build an off-the-shelf packages that will "fit" a reasonable number of in intermodal network design issues. Therefore logistics modeling languages appear to be the only avenue that will allow the necessary degree of tailoring and flexibility required for widespread success.

A modeling language consists of three fundamental pieces: (1) context specific data objects, (2) tools that operate on these objects, and (3) a macro language that can be used to reasonably quickly combine these objects and tools into custom systems. For example, the data objects used for spreadsheet modeling languages such as EXCEL are: cells, columns, and rows. Spreadsheet modeling languages include tools such as: select columns, insert column, sum data in a column, plot bar-chart, etc.

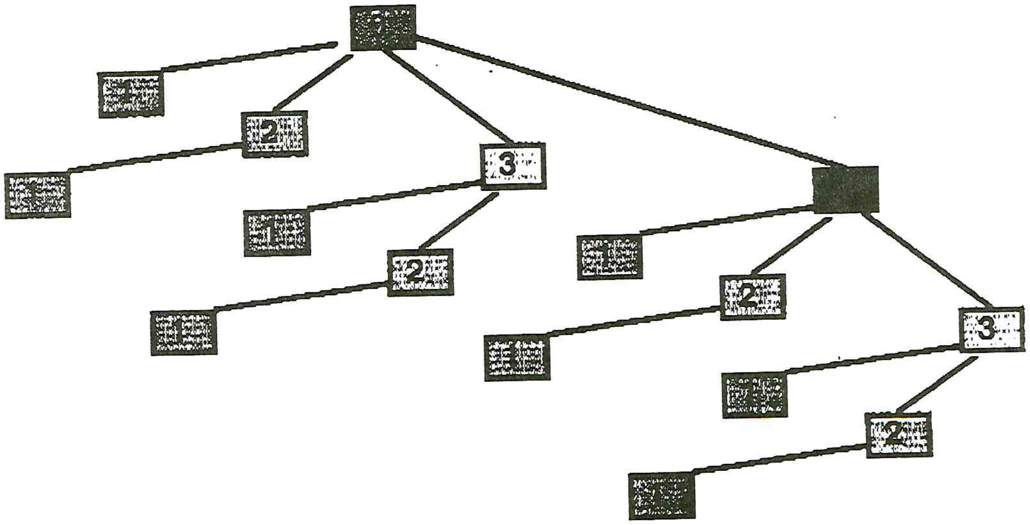
Logistics modeling languages to adequately address intermodal network design problems require a richer set of data objects such as maps, terminals, channels, networks, vehicles, routes, shipments, etc. Such languages also require a richer set of tools including draw map, construct network, find route, optimize flow, locate terminal, etc.

Optimization tools

Optimization concepts and algorithms as well as "optimization based" heuristics are critical to good intermodal network design. However, they must compliment rather than replace the ability of the human decision maker. The focus here will be on the how to develop the optimization elements so that they have both the speed and structure to be complimentary.

There are two places in the decision process where optimization is particularly important: the generation of alternative network configurations (e.g., Which ports should we use?) and the evaluation of each configuration (e.g., Which freight should flow through each port?). For some cases the evaluation problem can be adequately represented by a model that we can efficiently solve, such as a minimum cost flow problem. For these cases it is practical to at least consider optimization over the different configurations. When the evaluation problem is itself sufficiently difficult, then we are forced to consider only heuristics. In the latter case it is particularly important that the system have good interactive capability to allow the user to significantly influence the solutions considered.

Binomial tree search: Decisions regarding where to have terminals and how to link these terminals are naturally posed as combinatorial optimization problems. Most optimization approaches involve a partial "tree search" over the solution space. By organizing the decisions in a binomial tree, the user has greater control over the solutions being examined. A binomial tree search (illustrated in the figure below) uses an ordering of the decision variables to determine the best solution considering only variable {1} then the best solution using variables {1,2} etc. This is important when the user has some decision preference which is cannot easily be quantified and provides more information when the search is terminated before optimum.



Submodular set functions: An importation class of intermodal network design problems have an "independence" property among location decisions that allow them to be posed in terms of submodular set functions. This property allows some very efficient pruning of the binomial tree. It also allows the tree search to effectively utilize any solution that the user may have generated as a starting place. Finally, problems with this property allow a much more attractive class of heuristics than those without the property. These issues will be discussed together with their impact on computation.

Transportation for the 1996 Olympic Games

Design of the spectator transportation system for the 1996 Olympic Games in Atlanta provides an unusual and interesting intermodal network design problem. While perhaps an extreme case, it is illustrative of some of the practical problems that must be overcome in making our technology actually work. These issues together with the technology being used to address them will be discussed.

"INFORMATION TECHNOLOGY IMPLICATIONS FOR TRANSPORT OPERATIONS RESEARCH"

J.Barceló, Dept. d'Estadística i Investigació Operativa
Universitat Politècnica de Catalunya

The emerging applications of the Advanced Road Transport Telematic Technologies mainly address several operational areas: Demand Management, Travel and Information Systems, Integrated Urban Traffic Management Systems, Integrated Inter-Urban Traffic Management Systems, and so one. The development and implementation of such systems has prompted in recent years quite interesting research problems both from the modelling point of view and from the requirement of developing fast and efficient algorithms to solve these models.

To build these models a good understanding of the interactions that hold in a transportation systems is required, as well as a way of modelling them representing such interactions dynamically. Practical methods of measuring the dregree of change in network traffic flows, real-time identification of imbalance situations in the use of the available capacity, decision models to determine the adequate strategies, models to assess properly the impact of the management strategies, and so one, are only some examples of the above mentioned modelling and algorithmic developments required by systems using the new technologies.

Among such systems Travel and Information Systems, mainly in the domain of the in-car information systems, represent the paradigm of Telematic Application. The design and assessment of such systems is an interesting challenge for the operation researcher. How to estimate and forecast the evolution of network conditions, the dynamic identification of the best routes, the samplig procedures from floating car data resulting in the best estimates with less communications overheads, the assessment of the expected benefits, etc. are only a few examples of that assertion.

This paper will describe the general architecture of Travel and Information Systems, identifying the modelling and algorithmic requirements, and presenting a survey of the main developments in recent years, specially in the scope of the European Research and Development Programmes.

Recent Developments in Transportation Demand Modeling

Moshe Ben-Akiva
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Cambridge, USA

Emerging issues in transportation such as congestion pricing, telecommuting, traveler information systems, land use and environmental impacts require new developments in travel demand models which is the subject of this tutorial. The emphasis of the tutorial is on developments in disaggregate travel demand model systems in the last decade. Disaggregate models are calibrated on individual decision-maker data. These models explicitly take into account the choice processes the individual decision-maker undergoes that lead to travel demand.
Discrete choice models form the heart of such disaggregate model systems.

The tutorial will revolve around three themes:

1) Discrete choice models, especially the Multinomial Logit Model (MNL), have been widely used for the last 25 years. Motivated by the availability of cheaper and high powered computing, substantial developments have been made in discrete choice models, both in the econometric techniques and estimation methods adopted. Some of these developments include estimation methods for the Multinomial Probit Model and a new class of models known as Latent Class Choice Models.

2) The availability of alternative data sources for demand model estimation, such as revealed preferences and stated preferences, has catalyzed the development of econometric methods which explicitly capitalize on the advantages and correct for the disadvantages of alternative data sources. Specifically, combined Revealed Preference (RP) and Stated Preference (SP) model exploit advantages of both RP data and SP data, and improve on the accuracy of parameter estimates

3) Existing travel demand model systems do not capture adequately latent demand and timing decisions. They also have limited interdependencies among trip purposes, duration, mode, destination, etc. To overcome such deficiencies, an activity and travel model system which explicitly acoounts for such interpendencies will be presented.

PLENARY SESSIONS

Infinite Dimensional Variational Inequalities and Dynamic Network Disequilibrium Modeling

Terry L. Friesz
George Mason University

David H. Bernstein
Massachusetts Institute of Technology

Abstract

In this paper we explain the importance of modeling disequilibrium flow patterns occurring on networks, with special emphasis on automobile networks and the role of information technology. We show how elementary notions of disequilibrium, whether abstract, physical or economic in nature, give rise to an adjustment process expressible as a dynamical system. We comment that when such a system is autonomous its steady states can be given the traditional finite dimensional variational inequality/fixed point representations common to static network equilibria. Beyond this, and unique to our work, we show that if the disequilibrium dynamical system is nonautonomous it may tend toward moving or dynamic (instead of static) network equilibria expressible as infinite dimensional variational inequalities.

Using concepts of fast and slow dynamic systems, we show how day-to-day and within-day aspects of automobile travel decision making can be combined to yield a nonautonomous dynamical system with the mathematical properties reviewed previously. We introduce axioms for a proper predictive model of urban network flows which integrates both day-to-day and within-day considerations and postulate one such model for further study.

In particular we show that autonomous traffic network disequilibrium models based on extensions of the tatonnement paradigm of microeconomic theory are globally asymptotically stable for plausible regularity conditions using Lyapunov stability theory. Nonautonomous

versions of these models, reflecting combined departure time and route choice decisions, are evaluated to determine whether they produce trajectories which are attracted, not to single steady state, but to a bounded volume of the appropriate phase space using a stability theory based on average Lyapunov functions. Such stability, historically referred to as *stability in the sense of Lagrange*, is discussed with regard to its implications for traffic management. Notably such stability results admit the possibility of so-called *strange attractors*.

We then present the results of a variety of numerical experiments in which we search, using plausible effective cost operators and demand histories, for interesting dynamic phenomena. These experiments will not be completed before April 1994.

REAL-TIME ASSIGNMENT AND ROUTE GUIDANCE IN CONGESTED NETWORKS WITH MULTIPLE USER INFORMATION AVAILABILITY CLASSES

Hani S. Mahmassani
Department of Civil Engineering
The University of Texas at Austin
Austin, TX 78712, U.S.A.

Srinivas Peeta
Department of Civil Engineering
Purdue University
W. Lafayette, IN 47907, U.S.A.

Ta-Yin Hu
Athanasios Ziliaskopoulos
Department of Civil Engineering
The University of Texas at Austin
Austin, TX 78712
U.S.A.

Even under the most optimistic market penetration scenarios over the next decade, only a fraction of all vehicles in a network are expected to be equipped with in-vehicle electronic route guidance systems. Furthermore, equipped drivers may possess different information reception capabilities, or have access to different types of information. Drivers may also follow different behavior rules: equipped drivers may comply with prescribed or suggested routes, others will make their own decisions based on current or predicted conditions, and yet others may behave in a contrarian manner.

This paper addresses the problem faced by a central controller seeking to optimize overall network performance through the provision of real-time routing information to equipped motorists, taking into account different user classes in terms of information availability, information supply strategy, and response behavior. In particular, four user classes are incorporated in the formulation: 1) equipped drivers who follow prescribed system optimal routes; 2) equipped drivers who follow user optimum routes; 3) equipped drivers who follow a boundedly-rational switching rule in response to descriptive information on prevailing conditions; and 4) non-equipped drivers who follow externally specified paths, which may be historically known or solved for exogenously. However, the controller does not have a priori knowledge of the time-dependent O-D trip desires for users in each of these four classes over the whole duration of the planning horizon. Instead, the controller has reliable information of the demands only for a short interval into the future, and only historical information (or forecasts based on such information) for the remainder of the period of interest. For this reason, route assignmnet is performed on a quasi real-time basis, reflecting new information as it becomes available. A rolling horizon formulation and solution approach has been developed for this purpose.

Within the rolling horizon framework, the formulation seeks a time-dependent traffic assignment which provides the number of vehicles of each class on the network links and paths. While a solution is obtained for the whole planning stage, the routing information is implemented only for a short interval into the future (roll period); the problem is solved again for the next stage, starting at the end of the previous roll period. A simulation-based algorithm is presented to solve the problem at each stage, recognizing the interconnection between stages. The algorithm extends previous work by the authors for single-class time-dependent assignment. The DYNASMART simulation model is used to evaluate any particular path assignment pattern and provide the information necessary to guide the search to the solution satisfying the desired conditions. The interconnection between stages gives rise to unique challenges in terms of ensuring faithful simulation of the traffic system and correctly capturing-the dynamics of traffic routing through the network. The algorithm and associated simulation capabilities have been implemented in computer code and tested.

In addition to describing the formulation and solution procedure, the paper reports on the results of several numerical experimenst on test as well as actual traffic networks.

An Exact Algorithm for the Vehicle Routing Problem with Backhauls

Paolo Toth, Daniele Vigo

D.E.I.S. - Università di Bologna
Viale Risorgimento, 2 - 40136 Bologna - Italy

We consider the *Vehicle Routing Problem with Backhauls* (VRPB), also known as *Linehaul-Backhaul Problem*, as an extension of the Capacitated VRP in which the customer set is partitioned into two subsets. The first subset contains the Linehaul customers, each of which requires a given quantity of product to be delivered. The second subset contains the Backhaul customers, where a given quantity of inbound product must be picked up.

This customer partition is extremely frequent in practical situations. A common example is represented by the grocery industry, where supermarkets and shops are the Linehaul customers, and grocery suppliers are the Backhaul customers. In recent years it has been widely recognized that in this mixed distribution/collection context a significant saving in terms of transportation costs can be achieved by visiting Backhaul customers in distribution routes (see, e.g., Golden et al. [4]).

The VRPB then calls for determination of a set of vehicle routes visiting all customers such that: (i) for each route the total load associated to Linehaul and Backhaul customers does not exceed, separately, vehicle capacity; (ii) in each route the Backhaul customers are visited after all Linehaul customers; (iii) the total number of vehicles used and the total traveled distance are minimized. Precedence constraint (ii) is motivated by the fact that in many practical applications Linehaul customers have a higher priority. Moreover vehicles are often rearloaded, hence the on-board load rearrangement required by a "mixed" service is difficult, or even impossible, to carry out at customer locations.

More precisely, VPRB can be formulated as the following graph theory problem. Let $G' = (V', A')$ be a complete undirected graph, with vertex set $V' := \{0\} \cup$

$\{1,\ldots,n\} \cup \{n+1,\ldots,n+m\}$. Subsets $L = \{1,\ldots,n\}$, and $B = \{n+1,\ldots,n+m\}$, correspond to Linehaul and Backhaul customer sets, respectively. A nonnegative quantity, d_j, of product to be delivered or collected is associated with each vertex of $L \cup B$. Vertex 0 corresponds to the Depot (with a fictitious demand $d_0 = 0$), in which K identical vehicles with a given *capacity* D are stationed. Let c'_{ij} be the nonnegative *cost* associated with arc $(i,j) \in A'$, with $c'_{ij} = c'_{ji}$ for each $i,j \in V'$ such that $i \neq j$, and $c_{ii} = +\infty$ for each $i \in V'$. VRPB then consists of finding a min-cost collection of simple *circuits* (vehicle routes) such that:

(i) each circuit visits vertex 0;

(ii) each vertex $j \in V' \setminus \{0\}$ is visited by exactly one circuit;

(iii) the sum of the demands of the Linehaul and Backhaul vertices visited by a circuit does not exceed, separately, vehicle capacity, D;

(iv) in each circuit all deliveries must precede any pickup;

(v) each vehicle can perform at most one circuit;

(vi) the number of circuits (i.e. vehicles used) is minimum.

Observe that precedence constraint (iv) introduces an implicit orientation of the "mixed" vehicle routes, i.e. routes which visit both Linehaul and Backhaul customers. For the sake of simplicity in the following we assume, without loss of generality, that K_L, minimum number of vehicles needed to serve all Linehaul customers, is greater than or equal to K_B, minimum number of vehicles needed to serve all Backhaul customers. Indeed, where $K_L < K_B$, it is possible to solve the problem by building an equivalent instance obtained by exchanging subsets L and B. Because of the symmetry of the cost matrix, the solution to the original problem with $K_L < K_B$ can then be obtained by reversing orientation of the optimal vehicle routes. In view of requirements (v) and (vi), and in order to ensure feasibility, we also assume that K (number of available vehicles) is equal to the minimum number of vehicles needed to serve all customers. In our case we have $K = \max\{K_L, K_B\} = K_L$.

VRPB is known to be NP-hard (in the strong sense), since it generalizes the well-known *Capacitated Vehicle Routing Problem*, arising when $B = \emptyset$. Heuristic algorithms for the solution of VRPB have been proposed by Deif and Bodin [2], Casco, Golden and Wasil [1], Goetschalckx and Jacobs-Blecha [3], and Toth and Vigo [5]. An exact set-covering based algorithm for the special case of VRPB in which the number of customers of each type in a circuit is not greater than 4, has been proposed by Yano et al. [6].

In the following we present a new integer linear programming model for VRPB, by viewing it as an asymmetric problem. Let us define $L_0 := L \cup \{0\}$ and $B_0 := B \cup \{0\}$. Let $G = (V, A)$ be a directed graph obtained from G' by defining $V = V'$ and $A = \{(i,j) : i,j \in L_0\} \cup \{(i,j) : i \in L, j \in B\} \cup \{(i,j) : i \in B, j \in B_0\}$. A cost $c_{ij} = c'_{ij}$ (with $c_{ii} = +\infty$ for each $i \in V$) is associated with each arc $(i,j) \in A$. In other words this set contains:

- arcs from Linehaul vertices to Linehaul vertices and the depot, and vice-versa;
- arcs from Linehaul vertices to Backhaul vertices;
- arcs from Backhaul vertices to Backhaul vertices and the depot;

Indeed, note that no arc from a Backhaul to a Linehaul customer or from the depot to a Backhaul customer can belong to a feasible solution to VRPB, either because of the precedence constraint or the assumption $K_L \geq K_B$.

For each $S \subseteq L$ and each $S \subseteq B$, let $\sigma(S)$ be the minimum number of vehicles needed to serve all the customers in S, i.e. the optimal solution value of the *bin packing problem* with item set S and bin capacity equal to D. For each $i \in V$ let us also define $\Gamma_i^+ = \{j : (i,j) \in A\}$ (*forward star* of i) and $\Gamma_i^- = \{j : (j,i) \in A\}$ (*backward star* of i). A new integer linear programming formulation of VRPB is then:

$$(P) \qquad v(P) = \min \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{i \in \Gamma_j^-} x_{ij} = 1 \quad \text{for each } j \in V \setminus \{0\}; \tag{2}$$

$$\sum_{j \in \Gamma_i^+} x_{ij} = 1 \quad \text{for each } i \in V \setminus \{0\}; \tag{3}$$

$$\sum_{i \in \Gamma_0^-} x_{i0} = K_L; \tag{4}$$

$$\sum_{j \in \Gamma_0^+} x_{0j} = K_L; \tag{5}$$

$$\sum_{i\in S}\sum_{j\in\Gamma_i^+\setminus S} x_{ij} = \sum_{j\in S}\sum_{i\in\Gamma_j^-\setminus S} x_{ij} \geq \sigma(S) \qquad \begin{array}{ll}\text{for each} & S\subseteq L, S\neq\emptyset;\\ \text{and for each} & S\subseteq B, S\neq\emptyset;\end{array} \tag{6}$$

$$x_{ij}\in\{0,1\} \qquad \text{for each } i,j\in V; \tag{7}$$

where $x_{ij} = 1$ if and only if arc $(i,j) \in A$ is in the optimal solution. Equations (2)–(5) impose in-degree and out-degree constraints, while the so-called *capacity cut contraints* (6) impose both the connection and the capacity constraints.

We present a new lagrangian lower bound for VRPB based on projection of the feasible solutions space, which leads to determination of Shortest Spanning Arborescences with fixed in-degree or out-degree at the Depot. The lagrangian relaxation is then stregthened by adding valid inequalities, in a cutting-plane fashion. A branch-and-bound algorithm which makes use of reduction procedures, dominance criteria, feasibility checks, and heuristic algorithms is also presented. Extensive computational tests on several problem classes proposed in the literature show the effectiveness of the proposed approach.

References

[1] D.O. Casco, B.L. Golden, E.A. Wasil (1988). "Vehicle Routing with Backhauls: Models, Algorithms, and Case Studies", in *Vehicle Routing: Methods and Studies*, (B.L. Golden, A.A. Assad editors), North-Holland, Amsterdam, 127–147.

[2] I. Deif, L. Bodin (1984). "Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauling", in *Proceedings of the Babson Conference on Software Uses in Transportation and Logistic Management* (A.E. Kidder editor), Babson Park, 75–96.

[3] M. Goetschalckx, C. Jacobs-Blecha (1989). "The Vehicle Routing Problem with Backhauls", *European Journal of Operational Research* 42, 39–51.

[4] B.L. Golden, E. Baker, J. Alfaro e J. Schaffer (1985). "The Vehicle Routing Problem with Backhauling: Two Approaches" , in *Proceedings of the XXI Annual Meeting of S.E. TIMS* (R.D. Hammesfahr editor), Myrtle Beach, 90–92.

[5] P. Toth, D. Vigo (1992). "Heuristic Algorithms for the Vehicle Routing Problem with Backhauls", Research Report OR/92/7 DEIS, Bologna University

(presented at the 1st Meeting of the EURO Working Group on Urban Traffic and Transportation, Landshut, 1992).

[6] C. Yano, T. Chan, L. Richter, T. Cutler, K. Murty, D. McGettigan (1987). "Vehicle Routing at Quality Stores", *Interfaces* 17, 52–63.

A TABU SEARCH HEURISTIC FOR THE VEHICLE ROUTING PROBLEM WITH BACKHAULS

Michel Gendreau [(1)]
Alain Hertz [(2)]
Gilbert Laporte [(1)]

[(1)] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succ. A, Montréa, Québec, Canada H3C 3J7

[(2)] Département de Mathématiques, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

Extended abstract

The purpose of this paper is to propose a new heuristic method for the Vehicle Routing Problem with Backhauls (VRPB) defined as follows. Let G=(V,E) be a graph where $V=\{v_0,\ldots,v_n\}$ is the vertex set and $E=\{(v_i,v_j): i\neq j,\ v_i,v_j\in V\}$ is the edge set. We assume that all edges are undirected so that (v_i,v_j) is only defined for i<j. The vertex set is partitioned into $V=\{\{v_0\},L,B\}$ where v_0 is a depot at which *m* identical vehicles are based, L corresponds to linehaul customers and B to backhaul customers. A non-negative demand q_i is associated with each vertex v_i ($q_0=0$). With E is associated a cost matrix $C=(c_{ij})$ representing travel costs, distances or travel times. The VRPB consists of determining a set of *m* vehicle routes of least cost in such a way that

1) each route starts and ends at the depot
2) every vertex of $V\backslash\{v_0\}$ is visited exactly once by exactly one vehicle
3) the total demand of any route does not exceed the vehicle capacity Q
4) on any route, all backhaul customers are visited contiguously after all linehaul customers.

The Traveling Salesman Problem with Backhauls (TSPB) is a special case of the VRPB where only one vehicle (*m*=1) is available.

Relatively little previous research has been done on the development of good heuristics for the VRPB. Goetschalckx and Horsley (1986) have suggested an approach based on spacefilling curves, while Deif and Bodin (1984), Golden, Baker, Alfaro and Schaffer (1985) and Casco, Golden and Wasil (1988) use a modification of the Clarke and Wright savings algorithm to solve the VRPB. Groetschalckx and Jacobs-Blecha (1989) compare various basic heuristic strategies. The paper by Gélinas, Desrochers, Desrosiers and Solomon (1992) is different in that it describes an exact algorithm and the problem also incorporates time window constraints. In our opinion, the current state of knowledge on the VRPB is still unsatisfactory and more powerful algorithms should be designed. For example, when applied to the standard Vehicle Routing Problem (VRP), neither simple insertion procedures nor the Clarke and Wright algorithm rank among the best heuristics. Indeed, a recent comparison of the best known VRP heuristics has been done by the authors (1992) and it is shown that the Clarke and Wright heuristic is significantly worse than the best available algorithms.

We describe a tabu search procedure for the VRPB. It is based on three heuristic procedures developed by the authors for the TSPB and the VRP. The first procedure, GENI, is a generalized insertion routine. It is less myopic but more powerful than standard insertion procedures in that a vertex may only be inserted into a route containing one of its closest neighbors, and every insertion is executed simultaneously with a local reoptimization of the current tour. US is a post-optimization procedure that successively removes and reinserts every vertex, using GENI. The US procedure has produced highly satisfactory results on the TSP, better than Or-opt, for example. The combination of GENI and US yields a powerful two-phase heuristic for the TSP. These two procedures have been applied to the TSPB in the following way :

1) first, GENIUS is applied seperately to the vertices of L and to those of B

2) then, these two tours are combined in the following way. Consider $v' \in \{v_i,v_j\}$ and $v'' \in \{v_k,v_l\}$, where (v_i,v_j) is an edge of the first tour and (v_k,v_l) is an edge of the second tour. Let $w'=\{v_i,v_j\}\backslash\{v'\}$ and $w''=\{v_k,v_l\}\backslash\{v''\}$. Remove (v_i,v_j) and (v_k,v_l); introduce (v',v''), (v_0,w') and (v_0,w''). Select the least cost tour over all combinations $\{v',v''\}$.

This procedure has produced near-optimal solutions within modest execution times on randomly generated TSPB instances.

The best available algorithms for the VRP are all based on improvement methods such as simulated annealing and tabu search. These are search schemes in which successive neighbours of a solution are examined and the objective is allowed to deteriorate in order to avoid local minima. The TABUROUTE algorithm is such an algorithm. It has been developed by the authors in 1992 and is an adaptation of the tabu search technique to the VRP. The sequence of adjacent solutions is obtained by repeatedly removing a customer from its current route and reinserting it into another route. This is done by means of the procedures GENI and US.

We propose a new tabu search heuristic method for the VRPB which is based on a similar approach as TABUROUTE. At each iteration one customer v is moved from its current route r_1 to another route r_2. Customer v is inserted into r_2 by means of the method based on GENI and US for the TSPB. The reverse procedure is applied for the removal of v from r_1.

Numerical tests on a set of benchmark problems indicate that the new proposed procedure outperforms the best existing heuristic methods for the VRPB.

Zone Planning in Public Transportation Systems

by

Horst W. Hamacher*
and
Anita Schumacher*

Fachbereich Mathematik
and
Zentrum für Techno- und Wirtschaftsmathematik
Universität Kaiserslautern
Germany

1. Introduction:

In this paper we develop some tools for designing fair zones in public transportation networks. A *zone* in such a network is a set of stations which are treated as a unit as far as the fares for the passengers are concerned: The *zone tariff* is only dependent on the starting and ending zones of their travel. A *fair zoning* is one where the zone tariff is as close as possible to the *distance tariff* which relates the fare to the actual distance of a customer trip. In particular, the goal of a fair zone design is that neither the public transportation company nor the customer will have major disadvantages in the transition from distance tariff to zone tariff.

We will consider three objective functions which model this goal. If the zones are fixed we will show how to choose the tariffs in order to minimize the respective objective functions. This result will be combined with Greedy heuristics to design simultaneously zones and the corresponding tariffs. For graphs with special structures we give an integer programming formulation which is based on a connectivity property of the zones. The compexity status of the general zoning problem has as yet not been clarified, but it is shown that a closely related problem, the interior zoning problem, is NP hard.

2. Denotation:

We consider a public transportation system modeled by a graph $G=(V,E)$. The nodes $i\in V$ represent the stations and the edges $e=(i,j)\in E$ indicate that the two stations i and j are connected by at least one line between i and j. We assume that V has *n* nodes and m

* Partially supported by a grant of the Deutsche Forschungsgemeinschaft

edges. Each arc (i,j) has associated with it a nonnegative value c_{ij} representing the length of the direct connection between i and j. We assume that the graph G is connected such that the distance matrix $D = (d_{ij})$ is well-defined.

In the case of **distance tariffs** we assume that the ticket price for going from station i to station j in the public transportation network is proportional to d_{ij}. Although this type of tariff is a fair one, it is not very convenient: In order to find out a specific ticket price, the distance information needs to be available. This is feasible if the number of stations is small. Otherwise, this tariff system is too complex and not transparent enough for the customers.
Therefore, public transportation companies consider the introduction of **zone tariffs**: Within each of the zones a fixed ticket price is paid. For travel between zones a tariff is charged which is related to the "distance" of the zones instead of the distance between the stations. Obviously, we have to be more precise in describing what we mean by "zones" and "distance" of zones.

From a graph theoretical point of view, zoning is a partitioning of the node set V into disjoint node sets $V_1,\dots,V_L$. Subsequently, we call these subsets **zones**. We interpret each of the zones as node in the complete graph $G' = (V',E')$, i.e., $V' = \{V_1,\dots,V_L\}$ and $E' = \{ (V_l,V_k) : l,k =1,\dots,L\}$. (Notice that we include loops (V_l,V_l) in G'. They will be needed in our subsequent model.) If we define **zone costs** $c'(V_l,V_k)$ for all $l,k =1,\dots,L$ we can compute the LxL zone distance matrix

$$D' := (d'(V_l,V_k))$$

where $d'(V_l,V_k)$ is the length of a shortest path from V_l to V_k in G' with respect to edge costs c'. Since G' is a complete graph we may assume in the following that $c'(V_l,V_k) = d'(V_l,V_k)$ for all $l,k=1,\dots,L$ and that the matrix $C' = D'$ satisfies the triangle inequality

$$c'(V_l,V_k) \le c'(V_l,V_q) + c'(V_q,V_k) \quad \text{for all } l,q,k = 1,\dots,L \text{ with } l \neq k.$$

Zones and zone distances are used to represent distances between stations and thus to simplify the tariff system. In the **zone tariff** the ticket price for going from station $i \in V_l$ to station $j \in V_k$ with $i \neq j$ is given by $z_{ij} := d'(V_l,V_k)$. For i=j we define $z_{ii} := 0$. Obviously, the zone tariff is highly depending on the choice of the zones $V_1,\dots,V_L$ and the zone costs $c'(V_l,V_k)$.

3. Models for the Zone Design Problem

We propose the following three objective functions for measuring fairness in going from the distance tariff based on G to the zone tariff based on G':

(2.1) $$b_{max}(G,G') := \max_{i,j \in V} | d_{ij} - z_{ij} |$$

(2.2) $$b_1(G,G') := 1/2 \sum_{i,j \in V} | d_{ij} - z_{ij} |$$

(2.3) $$b_2(G,G') := 1/2 \sum_{i,j \in V} (d_{ij} - z_{ij})^2$$

Using the definition of objective functions (2.1) - (2.3) we can formulate the **zone design problem** as follows:

> Find a zone partitioning $V_1,...,V_L$ and costs $c'(V_l,V_k)$ such that $b(G,G')$ is minimum ($b \in \{b_{max}, b_1, b_2\}$).

The next result shows that we can concentrate on the zone partitioning of G, since the optimal zone costs are very simple to compute once we have chosen the zones.

Theorem 2.1:

Let $V_1,...,V_L$ be a zone partitioning of G.

In order to minimize $b_{max}(G,G')$, $b_1(G,G')$, and $b_2(G,G')$ we choose

a) $c'(V_l,V_k) =: c_{max}(V_l,V_k) := 1/2 \, (\max_{i \in V_l, j \in V_k} d_{ij} + \min_{i \in V_l, j \in V_k, i \neq j} d_{ij})$,

b) $c'(V_l,V_k) =: c_1(V_l,V_k) := \operatorname{median}_{i \in V_l, j \in V_k, i \neq j} d_{ij}$, and

c) $$c'(V_l,V_k) =: c_2(V_l,V_k) := \begin{cases} \dfrac{\sum_{i \in V_l, j \in V_k} d_{ij}}{|V_l| \cdot |V_k|} & for\ l \neq k \\ \dfrac{2 \cdot \sum_{i,j \in V_l} d_{ij}}{|V_l| \cdot (|V_l| - 1)} & for\ l = k \end{cases}$$,respectively.

An immediate consequence of Theorem 2.1 is the following result.

Corollary 2.2:

Let $V_1,\dots,V_L$ be a zone partitioning of G and let $c'(V_l,V_k)$ be any zone cost. Then

a) $b_{max}(G,G') \geq 1/2 \max_{l,k=1,\dots,L\,,\,|V_l|\,|V_k|>1} \{ \max_{i\in V_l,\, j\in V_k} d_{ij} - \min_{i\in V_l,\, j\in V_k,\, i\neq j} d_{ij} \}$

b) Let

$V_{kl}^{+} := \{d_{ij} : i\in V_k,\, j\in V_l,\, d_{ij} \geq \operatorname{median}_{i\in V_l, j\in V_k} d_{ij} \}$ and

$V_{kl}^{-} := \{d_{ij} : i\in V_k,\, j\in V_l,\, d_{ij} \leq \operatorname{median}_{i\in V_l, j\in V_k} d_{ij} \}$. Then

$$b_1(G,G') \geq 1/2 \sum_{l,k=1,\dots,L} (\sum_{d\in V_{kl}^{+}} d_{ij} - \sum_{d\in V_{kl}^{-}} d_{ij})$$

c) $b_2(G,G') \geq 1/2 \sum_{l,k=1,\dots,L} Var\{d_{ij} : i\in V_l,\, j\in V_k,\, i\neq j\}$

All inequalities hold with equality if the zone costs are chosen according to Theorem 2.1, respectively.

4. Greedy Heuristic for the Zone Design Problem

Greedy heuristics start with n zones, where each zone corresponds to a single node and combine iteratively two zones to a new one. After updating the graph this process is continued until the planned number of zones is attained.

The goal in each iteration is to achieve a minimal increase in the objective function b(G,G'). For this purpose we combine two zones X and Y which yield a minimal increase f(X,Y) in the objective function. The definition of the function f(X,Y) is depending on the choice of the objective function b(G,G').

Depending on the chosen objective introduced in Section 3 different strategies for good functions f(X,Y) are used in the resulting versions of the Greedy algorithm. We applied these heuristics to data of a German transportation company and compared them with clustering algorithms from literature. The results indicate that the Greedy approach is preferable to the latter algorithms. We will report on these results.

5. Complexity Issues

As yet, the complexity status of the general zone planning problem is not clear. For the case where only the interior zoning tariffs are considered, i.e. the deviation of distance and zone tariffs only within each of the zones, it is proved that the problem of finding a fair zone planning is NP-hard for all three choices of objective functions introduced in Section 2.

6. Integer Programming Formulation

We consider binary decision variables h_{kl} defined by $h_{kl}=1$ iff edge k of G is contained in zone l. In the case where the graph G is a line graph, we can prove connectivity properties of the optimal zone planning which lead to an integer programming formulation of the zone planning problem using appropriately defined cost vectors c^{ij}:

$$\min \{\max \{c^{ij} h : i<j \} \}$$

subject to

$$\sum_{k=1}^{m} h_{lk} \le 1 \; for\ all\ l$$

$$\sum_{l=1}^{m} \sum_{k=1}^{m} h_{lk} \ge n - m$$

$$h_{ij} + h_{kj} - h_{i+1,j} \le 1$$

Consequences of this IP formulation including exact solution approaches will be discussed.

7. Conclusion and Further Research

While the zone tariff presented in this paper is far less complex than the distance tariff, it may nevertheless be desirable to simplify it even further by requiring that the ticket price is only depending on the number of crossed zones, independent on where these zones are located. For this problem the computation of approximate solutions by good heuristics and the derivation of exact algorithms are under research.

The models of this paper neglect two important issues: The frequency of usage and the impact of tariff changes on this frequence. While the former can be incorporated into our model by adding multipliers m_{ij} to each of the terms $|d_{ij} - z_{ij}|$ and $(d_{ij} - z_{ij})^2$, respectively, the consideration of passenger reactions to tariff changes will require a completely different model. It should be noted however that the goal of our approach is to minimze the change in ticket prices such that this effect is avoided as much as possible.

Regional mass transit assignment with resource constraints(*)

Paolo Carraresi(†), *Federico Malucelli, Stefano Pallottino*(**)

Abstract

A mass transit system in a regional area is considered. In this paper the problem of improving the quality of the service considering passengers assignment is studied. A model to improve the effectiveness of the system without worsening the cost (vehicles and duties) and the traffic assignment is proposed. A new definition of passenger assignment to support the proposed model is given.

1. Introduction

In this paper we will consider a mass transit system in a regional area where possibly both road and rail transportation may be used by commuters. Unlike the urban situation [10], in a regional mass transit system line frequencies are relatively low and trips can be considered as singular entities. We will assume that all the trips run on time according to a given time-table which is known by users; moreover, the vehicles have a fixed capacity (number of seats) which cannot be exceeded. In the literature, the problem of choosing jointly departure time and route in a mass transit system has been studied without explicitly considering the capacity constraints, hence without considering the deriving flow priority requirements [2, 6, 11-13].

Assuming that an assignment of passengers to vehicles is known, the first problem we will consider is to improve the effectiveness of the mass transit system. To this aim, a model which minimizes passengers total waiting time by modifying the departure times of the trips is proposed. Constraints are imposed in such a way that changing the departure time of the trips will maintain the feasibility of vehicles and drivers scheduling, and will guarantee that the starting assignment is still proper. For these reasons, such kind of model is called *conservative* and must be considered as first step towards the study of a general framework where resource constraints are explicitly handled and the goal is to obtain a new assignment perceived by the users to be not worse than the previous one. Under quite reasonable hypotheses, the conservative problem can be solved by means of a network flow algorithm, while considering the general case it turns out that the problem be modelled as a Quadratic Semi-Assignment.

The second problem we consider is to define assignments which are suitable inputs for the conservative model previously introduced. Here we will propose two alternative definitions of assignment: in the first one the capacity constraints are taken into account explicitly and a so called ε-feasible flow is considered [11, 12], while in the second one these constraints are substituted by continuous penalty functions in order to find a user equilibrium assignment.

(*) This work has been supported by grants n. 93.01799.PF74, 93.01669.PF74 of Progetto Finalizzato Trasporti Due, Italian National Research Council (CNR).

(†) Professor Paolo Carraresi died unexpectedly on March 5, 1994. At the time of his death this paper was substantially completed. Doing the final revision, the other two authors found inspiration in the suggestions, ideas and all they could learn after so many years of common work.

(**) Department of Computer Science, University of Pisa, 40, Corso Italia, I-56125 Pisa, Italy

The paper is organized as follows. In section 2 the transit service is described and the basic notation is introduced. The notion of assignment is presented in section 3. Section 4 is devoted to the conservative model to improve of the effectiveness of the transit system. In section 5 new definitions of assignment are discussed. The final section contains some concluding remarks.

2. The regional mass transit service

Let us focus now on the main characteristics of the service perceived by the users.

The service is described in terms of both bus and rail lines. A *line* connects a sequence of geographical points, called *stops*, where passengers can board and alight. At least one trip is associated to each line.

A *trip* is a run of a vehicle from the starting terminal to the ending terminal of a line according to a time-table which gives the arrival and departure times at each stop of the line. We assume that time-tables are known by passengers and that the service is *regular*, i.e. vehicles are on schedule at each stop. Vehicles have a fixed capacity, usually given by the number of seats.

The *demand* is located in ideal geographical points, called *centroids*, and it is described in terms of passengers travelling between two centroids, called *origin* and *destination*. The demand for each origin/destination (o/d) pair is partitioned into *groups*. The passengers of each group have common socio-economic conditions and share either a common *target time* at the destination or a common *target time* at the origin. A target time at the destination means that a passenger does not want to arrive later than that time, while a target time at the origin means that a passenger does not want to leave the origin before that time. In the following, we assume that the demand is known, that is the number of passengers for each o/d pair and for each group is given.

A regional mass transit service can be described by two networks: a *sketch* network and a *space-time* network.

The nodes of the sketch network represent centroids or stops while the arcs can be distinguished in *in-vehicle arcs*, connecting two consecutive stops on a line, and *walking arcs*, connecting, in both directions, centroids and neighboring stops as well as stops which are located within walking distance.

On the other hand, the space-time network (V,E) describes the transit service by representing each trip separately. In practice, the space-time network does not describe only the lines structure, but it considers also how trips unroll during the time.

Each node u of V corresponds to a pair $\omega(u)=[i,t_i]$ where i is a location (stop or centroid) and t_i is a time; u refers to an arrival to, or a departure from, location i at time t_i.

In fig. 1, a sketch network and the corresponding space-time network is reported. The sketch network describes a transit system with one origin o and two destinations d_1 and d_2 three lines (a, b and c), five stops and four walking arcs one of which connects stop 2 with stop 5, while the space-time network describes one trip for each line.

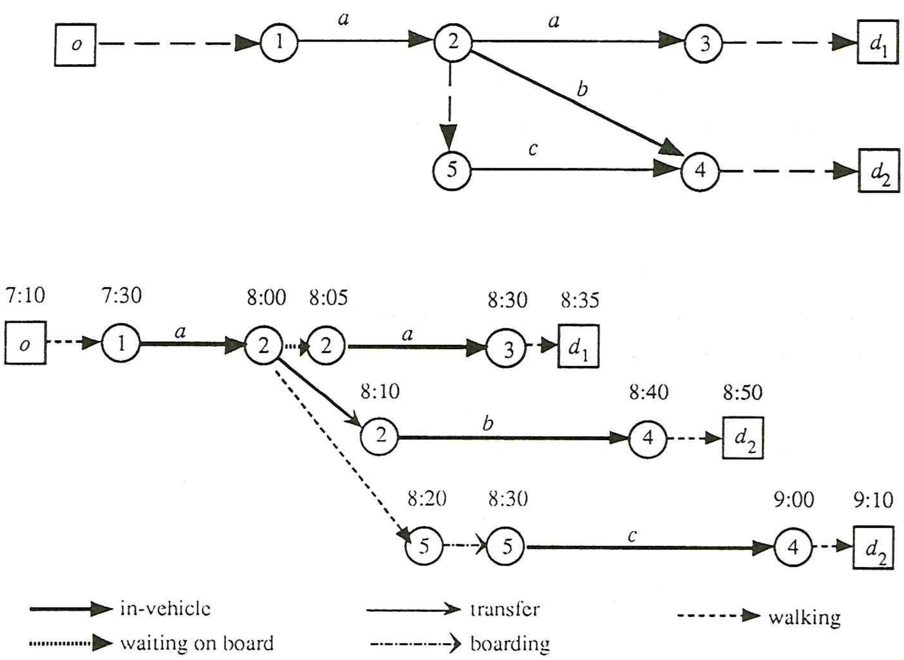


fig. 1

The set of arcs E can be partitioned as follows.

- *In-vehicle arcs E_v.*

  An in-vehicle arc (u,v), where $\omega(u)= [i,t_i]$ and $\omega(v) = [j,t_j]$, represents a portion of a trip starting at location i at time t_i and arriving at the next location j at time t_j (e.g. ([1, 7:30],[2, 8:00]) in fig. 1). A capacity u_e is given for each $e \in E_v$. This capacity depends on the vehicle running the corresponding trip.

- *Walking arcs E_w.*

  Walking arcs allow passengers to reach a stop starting from a centroid or another stop and vice versa. In a walking arc (u,v), $\omega(u)= [i,t_i]$, $\omega(v) = [j,t_j]$, where i, j represent stops or centroids and t_j - t_i is the walking time to get location j starting from location i at t_i (e.g. ([2, 8:00],[5, 8:20]) in fig. 1).

- *Boarding and alighting arcs E_a.*

  Boarding and alighting arcs represent passenger ingress to and egress from a vehicle, respectively. A boarding arc (u,v) connects the ending node of a walking arc to the starting node of an in-vehicle arc. The locations referred by u and v are coincident, that is $\omega(u)= [i, t_i]$ and $\omega(v) = [i, t_i]$ and t'_i - t_i represents the waiting time at the stop i (e.g. ([5, 8:20],[5, 8:30]) in fig. 1). An alighting arc connects the ending node of an in-vehicle arc to the starting node of a walking arc. Usually no time penalty is associated to passenger egress. In such a case $\omega(u)=\omega(v)$ which implies that the alighting arc is considered together with the walking arc (e.g. ([2, 8:00],[5, 8:20]) in fig. 1).

- *Transfer arcs* E_t.

  Transfer arcs connect nodes of different in-vehicle arcs and represent passenger transfer between trips at a given stop. Obviously, a transfer arc (u,v) at stop i, $\omega(u)= [i,t_i]$, $\omega(v) =[i, t'_i]$ exists if the waiting time $t'_i - t_i$ allows to transfer (e.g. ([2, 8:00],[2, 8:10]) in fig. 1).

- *Waiting on board* E_b.

  A waiting on board arc (u,v) connects two consecutive in-vehicle arcs related to the same trip; u and v refer to the same stop i, that is $\omega(u)= [i, t_i]$ and $\omega(v) = [i, t_i]$, and $t'_i - t_i$ represents the waiting time on board at stop i (e.g. ([2, 8:00],[2, 8:05]) in fig. 1).

Note that the space-time network is an acyclic graph.

3. Passenger assignment and ε-feasible flow

In the following we will show that a passenger assignment can be described as a multicommodity feasible flow on (V,E). Commodities correspond to groups of passengers with the same o/d pair, sharing either a common target time τ_d at the destination d or a common target time τ_o at the origin o. We assume that for each group k =1,..., K, only one of the two target times defines a constraint. Hence in the pair (τ_o,τ_d) either $\tau_o =0$ or $\tau_d =+\infty$.

A *feasible o/d path* for the group k is a path on (V,E) from u to v, where $\omega(u)= [o,t'_o]$, $\omega(v) = [d, t'_d]$, o and d correspond to the origin and destination centroids of the group, $t'_o \geq \tau_o$ and $t'_d \leq \tau_d$.

Let P_k denote the *set of feasible o/d paths* for commodity, and D_k be the demand of commodity k (i.e. the number of passengers of the group corresponding to commodity k) k=1,..., K. Moreover, given any path p we will denote the number of passengers using p by h_p. Finally let δ_{ep}=1 if the arc e belongs to the path p, and 0 otherwise.

Define a *passenger assignment* as a feasible multicommodity flow $h =[h_p]$, $p\in P_k$, k =1,..., K, on (V,E), satisfying the demand and the arc capacities; the set F of all feasible multicommodity flows can be described as follows:

$$
\begin{array}{lll}
F =\{h: & & \\
\quad \sum\limits_{p\in P_k} h_p=D_k, & k=1,\ldots, K, & \\
\quad \sum\limits_{k=1}^{K} \sum\limits_{p\in P_k} h_p\delta_{ep} \leq u_e, & \forall\ e\in E_v, & (3.1)\\
\quad h_p\geq 0, & \forall\ p\in P_k,\ k=1,\ldots,K\}. &
\end{array}
$$

We assume that passengers of the same group behave in the same way when selecting a feasible o/d path. Group behavior depends on a *perceived or generalized cost* of o/d paths.

The generalized cost of a path is a weighted function of the following components: in-vehicle time, waiting time, walking distance, departure tardiness, arrival earliness, number of transfers, trip fare, etc. Here we will consider two kinds of generalized cost.

3.1 First generalized cost: weighted sum of trip time, waiting time and tardiness/earliness

The first kind of generalized cost, quite simple and widely used [4, 5, 9, 10], is the weighted sum of trip time, waiting time and tardiness/earliness with respect to the target times. This cost has the following two components:

i) the difference between the target time and the departure time at the origin, if the target time refers to the destination; or, symmetrically, the difference between the arrival time and the target time, if the target time refers to the origin;

ii) the total waiting time spent in transfers.

Let c_p be the *generalized cost* of path p and let γ_k be the cost of the minimum cost path for the commodity k, that is $\gamma_k = \min\{c_p: p \in P_k\}$, $k=1,\dots, K$.

We assume that passengers of the same group perceive as equivalent two feasible o/d paths p and q, $c_p \geq c_q$, when the relative difference of the costs, $(c_p - c_q)/c_q$, is not greater than a given relative tolerance $\varepsilon>0$. We will assume that passengers of the same group select, among all feasible paths with available capacity, that minimizing the generalized cost within the given tolerance ε.

We can define now an ε-feasible flow as an assignment of all passengers to perceived minimum cost paths only.

Definition 1

A passenger assignment h is an *ε-feasible flow* if $h \in F$ and *the relative cost* $(c_p - \gamma_k)/\gamma_k$ of each path p used by passengers of commodity k is less than or equal to ε.

Let us denote by $P_k(\varepsilon)$ the *set of all ε-feasible paths* for commodity k, that is paths which are perceived as equivalent to the minimum cost path:

$$P_k(\varepsilon)=\{p \in P_k: c_p \leq \gamma_k (1+\varepsilon)\}. \tag{3.2}$$

3.2 Second generalized cost: differences with the ideal departure, restarting and arrival times

Here we consider a simplified passenger behavioral model. We define the *ideal departure, restarting and arriving times* as the best times a passenger can select at any moment regardless the actual seat availability on the vehicles. For example, consider a group of passengers having target time at the destination: for this group of passengers the ideal departure time is the starting time of the latest trip which allows them to arrive at the destination within the target time. Consider now a group of passengers which arrives at a transfer place: for this group of passengers and the considered stop, the ideal restarting time is the earliest departure time of a trip which allows them to reach their destination within the target time.

For the groups of passengers having target time at the origin, the ideal arriving time is a reasonable time they could expect to arrive at destination starting form the origin not before the target time. The ideal departure time is the starting time of the earliest trip which allows passengers to arrive at destination not later than the ideal arriving time. The ideal restarting time at each transfer place is the starting time of the earliest trip which allows passengers to arrive at destination not later than the ideal arriving time.

Hence, for each group of passengers, we can associate to each node of a feasible path (origin, transfer stops or destination) a value given by the difference with the ideal choice.

On the basis of the above definitions, we assume that passengers behave according to the following rules:

- *latest departure time* for groups of passengers with target time at the destination: no passenger will leave the origin earlier than necessary in order to reach the destination no later than the target time;
- *earliest departure time* for group of passengers with target time at the origin: no passenger will deliberately delay the starting time at the origin;
- *earliest restarting time* for all the groups: at any transfer place no passenger will deliberately delay the restarting time.

This model has been proposed in [11] where it is shown that these behavioral assumptions can be transformed into generalized costs of o/d paths.

Assuming this behavioral model, we define the indifference tolerance ε as the maximum difference with the ideal times at the origin, at the destination and at each transfer stop of the o/d path. Note that, unlikely the other definition of ε tolerance, in this case ε is an absolute value.

As we did in the case of the first kind of generalized cost, given a tolerance ε, $P_k(\varepsilon)$ denotes the set of paths perceived equivalent by passengers of commodity k , that is the set of paths whose starting times at the origin, at the transfer stops and at the destination differ with the ideal times by less than ε.

3.3 Mathematical formulation

Let us suppose that a tolerance $\varepsilon>0$ is given; this tolerance is a relative value if the first generalized cost is considered, while it is an absolute value if the second kind of generalized cost is used. It is well known that this value is difficult to estimate and it is somewhat arbitrary; in fact that tolerance is closely related to the quality of service that passengers perceive and to their travel experience. Nevertheless, for this moment we consider ε as a value which really represents the threshold of the indifference on the generalized costs; later, in section 5 we will characterize this threshold.

Definition 2

A passenger assignment h is an ε-feasible flow if $h\in F$ and $h_p=0$ if $p\in P_k \backslash P_k(\varepsilon)$.

By $F(\varepsilon)$ we will denote the *set of all ε-feasible flows*:

$$
\begin{aligned}
F(\varepsilon)=\{h:\\
&\sum_{p\in P_k(\varepsilon)} h_p=D_k, && k=1,\dots, K,\\
&\sum_{k=1}^{K}\sum_{p\in P_k(\varepsilon)} h_p\delta_{ep} \le u_e, && \forall\ e\in E_v,\\
&h_p\ge 0, && \forall\ p\in P_k(\varepsilon),\ k=1,\dots, K\}.
\end{aligned}
\tag{3.3}
$$

Hence, an assignment $h\in F(\varepsilon)$ can be obtained by solving a linear system where the number of variables is the number of ε-feasible paths. This means that determining whether $F(\varepsilon)=\emptyset$ or a feasible flow $h\in F(\varepsilon)$ exists can be done solving problem (3.3). As in real applications the number of ε-feasible paths is order of the cardinality of the set of arcs, solving (3.3) is a relatively easy task.

4. Improving the effectiveness of the transit system: a conservative model

In this section we introduce a time-table design model which consider the definition of generalized cost and the passenger behavior based on the earliest/latest departure times and earliest restarting times rules (see section 3.2).

In the following, we assume that an ε-feasible flow is given together with the vehicle and drivers' scheduling. A *vehicle scheduling* is specified by blocks, where a *block* is the sequence of trips consecutively run by one vehicle. A *drivers' scheduling* is a set of driver duties, where a *duty* is a set of pieces of work satisfying union regulations. A *piece of work* is a continuous driving period between two relief places, that is places where a driver substitution can occur. For more details on vehicle and drivers' scheduling see [3]. Without loss of generality, in the following we assume that there is at most one relief point between two trips.

Our aim is to modify the trip departure times in order to minimize the total generalized cost in such a way that the given assignment is still an ε-feasible flow and the given vehicle and driver scheduling are still feasible.

Now consider the problem of improving the effectiveness of the transit system. The problem can be represented by means of a directed graph $G=(N,A)$. Each node in $N=\{1,\dots,n\}$ corresponds to a trip, and the arc set is defined as the union of the following set of arcs (i.e. $A=A_1\cup A_2\cup A_3\cup A_4$).

A_1: there is an arc $(i,j)\in A_1$ iff there exists a non empty set of passengers transferring from trip i to trip j, possibly also walking between two stops.

A_2: there is an arc $(i,j)\in A_2$ iff there exists a stop where a group of passengers splits between trips i and j. In other words portions of trips i and j belong to alternative ε-feasible paths used by the same group.

A_3: there is an arc $(i,j)\in A_3$ iff trips i and j are consecutive within the same vehicle block.

A_4: there is an arc $(i,j)\in A_4$ iff there exists at least one duty where a piece of work on trip i is followed by a piece of work on trip j.

Note that A_1 and A_2 are defined by the ε-feasible flow on (V, E), while A_3 and A_4 depend on vehicle and drivers' scheduling, respectively.

Now consider to modify the departure times of the trips. Let π_i denote the amount of variation related to trip i. Note that variables π_i are not restricted in sign: $\pi_i<0$ means that trip i is anticipated, while $\pi_i>0$ means that trip i is delayed and $\pi_i=0$ does not modify the current timetable.

Further on, we will show that the problem of minimizing the total generalized cost in such a way that the given assignment is still an ε-feasible flow and the given vehicle and drivers' scheduling are still feasible can be formulated as the following LP problem:

$$\begin{array}{lll} \min & \sum_{i} c_i\pi_i & \\ & \pi_j\text{-}\pi_i \geq s_{ij} & \forall\ (i,j)\in A, \\ & l_i \leq \pi_i \leq u_i & \forall\ i\in N. \end{array} \qquad (4.1)$$

Let us examine in detail the problem constraints.

Consider $(i,j)\in A_1$, that is there is at least one group of passengers which transfer from trip i to trip j at a given transfer stop. Let us denote by a_i the arrival time of trip i, d_j the departure time of trip j, τ_{ij} the transfer time (possibly including the walking time); then the waiting time is equal to $\text{-}s^1_{ij}$ where $s^1_{ij} = a_i + \tau_{ij} - d_j \leq 0$.

To maintain the possibility of transfer from trip i to trip j, the new departure and arrival times $a'_i=a_i+\pi_i$ and $d'_j=d_j+\pi_j$ must satisfy the condition $a'_i+ \tau_{ij} \leq d'_j$. This implies the following constraints:

$$\pi_j\text{-}\pi_i\geq s^1_{ij} \qquad \forall\ (i,j)\in A_1.$$

Consider $(i,j)\in A_2$. One case is when there is at least one group of passengers which, at a given node (either the origin or a transfer stop) board two different trips i and j whose departure times d_i and d_j are such that $\text{-}\varepsilon \leq d_j - d_i \leq \varepsilon$. Note that if $(i,j)\in A_2$, also $(j,i)\in A_2$. In order to maintain the ε-feasibility of passengers assignment, the new departure times must satisfy the same conditions; that is

$$\text{-}\varepsilon + d_i - d_j \leq \pi_j - \pi_i \leq \varepsilon + d_i \text{-}d_j.$$

If we denote by $s^2_{ij} = \text{-}\varepsilon + d_i - d_j$ (≤ 0) and by $s^2_{ji}= \text{-}\varepsilon + d_j \text{-}d_i$ (≤ 0), we obtain the following constraints:

$$\pi_j\text{-}\pi_i\geq s^2_{ij} \qquad \forall\ (i,j)\in A_2,$$

$$\pi_i\text{-}\pi_j\geq s^2_{ji} \qquad \forall\ (j,i)\in A_2.$$

There is another case which induces a pair of arcs (i,j) and $(j,i)\in A_2$. Consider a group of passengers with target time at the origin which splits on different paths; assume that there are at least two different paths using final trips i and j, respectively, to reach the destination. According to the ε-feasibility we have $\text{-}\varepsilon \leq a_j - a_i \leq \varepsilon$. In analogous way to the previous case, defining

$s_{ij}^2 = -\varepsilon + a_i - a_j$ (≤ 0) and $s_{ji}^2 = -\varepsilon + a_j - a_i$ (≤ 0), in order to maintain the ε-feasibility we must impose:

$$\pi_j - \pi_i \geq s_{ij}^2 \qquad \forall\ (i,j) \in A_2,$$

$$\pi_i - \pi_j \geq s_{ji}^2 \qquad \forall\ (j,i) \in A_2.$$

Consider $(i,j) \in A_3$, that is trip j follows immediately trip i in the scheduling of a vehicle. This means that $a_i + \delta_{ij} \leq d_j$, where a_i is the arrival time at the ending terminal of trip i, d_j is the departure time from the starting terminal of trip j and δ_{ij} is the time due to the deadheading trip between the two terminals. If we denote by $s_{ij}^3 = \delta_{ij} + a_i - d_j$ (≤ 0), the scheduling remains feasible if the following constraints hold:

$$\pi_j - \pi_i \geq s_{ij}^3 \qquad \forall\ (i,j) \in A_3.$$

Consider $(i,j) \in A_4$, that is there is one duty where a piece of work on trip j follows immediately a piece of work on trip j. This means that $b_i + \rho_{ij} \leq g_j$, where b_i is the ending time of the piece of work on trip i, g_j is the starting time of the piece of work on trip j, and ρ_{ij} is a spread time between the two pieces of work. This time depends on possible union regulations and on the time needed to move between the relief points of trips i and j. If we denote by $s_{ij}^4 = b_i + \rho_{ij} - g_j$ (≤ 0), to maintain the feasibility of duties the following constraints must hold:

$$\pi_j - \pi_i \geq s_{ij}^4 \qquad \forall\ (i,j) \in A_4.$$

Finally let

$$s_{ij} = \max\{s_{ij}^r, (i,j) \in A_r, r=1,\ldots,4\} \qquad \forall\ (i,j) \in A.$$

Time window constraints of problem (4.1) (i.e. $l_i \leq \pi_i \leq u_i$, $\forall i \in N$) originate from target times at the origin and/or destination for each group of passengers.

Firstly define l_i, $i \in N$. Let G_i be the set of groups of passengers k having target time at the origin and beginning the travel by boarding trip i. Then l_i^k is the smallest π_i which allows passengers of group k to leave the origin not before the target time and to board trip i (possibly after walking to the stop). Then l_i is given by

$$l_i = \max\{l_i^k, k \in G_i\} \qquad \forall\ i \in N.$$

Analogously we can define u_i, $i \in N$. Let G'_i be the set of groups of passengers k having target time at the destination and ending the travel by alighting trip i. Then u_i^k is the largest π_i which allows passengers of group k to alight trip i and reach the destination not later than the target time (possibly after walking to the destination). Then u_i is given by

$$u_i = \min\{u_i^k, k \in G'_i\} \qquad \forall\ i \in N.$$

Now let us detail the objective function min $\Sigma_i\, c_i \pi_i$. Consider trip i. The coefficient c_i is a weighted sum of the number of passengers boarding trip i whose travel cost is affected by a variation on the departure time of trip i. Let us denote by α and β suitable non negative coefficients that measure the perception of the gap with the ideal departure time at the origin and with the ideal arriving time at the destination (α) and the gap with the ideal waiting time at transfer stops (β).

Let us consider separately the set of groups of passengers having target time at the origin and travelling on trip i, K_i^o, and the set of groups of passengers having target time at the destination and travelling on trip i, K_i^d. The perception of the variation π_i depends whether passengers board trip i coming from the origin, or coming from another trip, or alight trip i to reach the destination, or to transfer to another trip.

When a passenger of group k boards trip i coming from the origin, he/she perceives a variation $\alpha\pi_i$ in his/her travel cost if $k \in K_i^o$, while he/she perceives a variation $-\alpha\pi_i$ in his/her travel cost if $k \in K_i^d$. In fact in the first case, according to the earliest departure time rule, the gap with the ideal departure time at the origin increases with π_i while in the second case, according to the latest departure time rule, this gap decreases when π_i increases. When a passenger of group k boards trip i coming from another trip, he/she perceives a variation $\beta\pi_i$ in his/her travel cost; in fact the waiting time for trip i increases with π_i.

When a passenger of group k alights trip i going to another trip, he/she perceives a variation $-\beta\pi_i$ in his/her travel cost; in fact the waiting time for the following trip decreases when π_i increases.

When a passenger of group k alights trip i going to the destination, he/she perceives a variation $\alpha\pi_i$ in his/her travel cost if $k \in K_i^o$, while he/she does not perceive any variation if $k \in K_i^d$. In fact in the first case, the gap with the ideal arrival time at the destination increases with π_i, while in the second case, the variation π_i does not affect the travel cost as for this group of passengers it is sufficient to reach the destination within the target time.

Hence denoting by:

- H_{ij}^k the number of passengers of group k transferring from trip i to trip j,
- H_{oi}^k the number of passengers of group k boarding trip i coming from the origin,
- H_{id}^k the number of passengers of group k alighting trip i to reach the destination,

coefficient c_i is given by:

$$c_i = \alpha \left(\sum_{k \in K_i^o} (H_{oi}^k + H_{id}^k) - \sum_{k \in K_i^d} H_{oi}^k \right) + \beta \sum_{j \neq i} \sum_{k \in K_i^o \cup K_i^d} (H_{ji}^k - H_{ij}^k).$$

Consequently, given an assignment h, a cost c_i can be computed for each trip $i \in N$. According to the definition of cost c_i, solving (4.1) provides trip departure variations π_i such that:

- the starting assignment h is an ε-feasible flow;
- the given vehicle blocks and driver duties modified according to π, are still feasible;
- the generalized cost of assignment h is minimized.

Note that problem (4.1) is the dual of the following max cost flow problem :

$$\max \sum_{ij} s_{ij} x_{ij} + \sum_{i} l_i y_i - \sum_{i} u_i z_i$$
$$Ex + y - z = c$$
$$x, y, z \geq 0,$$

where E denotes the node-arc incidence matrix of graph G. Hence problem (4.1) can be solved in polynomial time [1].

If we impose further restrictions to the values of π_i, model (4.1) is no more suitable and we must introduce a more general one. As a first example of restrictions consider the case where trip departure time variations must belong to finite sets of discrete values which can vary depending on the trip (e.g. $\pi_i \in \{-10, -5, 0, 5, 10\}$, $\pi_j \in \{-60, -45, -30, -15, 0, 15\}$). These requirements may occur at main stops, where trip departure at given times is preferred by passengers.

A second example takes into account more general equilibrium constraints. In model (4.1) constraints on pairs of trips in A_2 consider "local properties" of the ε-feasible based on the earliest restarting time rule. If the generalized cost of the path refers to the whole path and not only to the local properties at a stop, the constraints on the given ε-feasible flow h can be expressed as follows:

$$c_p(\pi) - \gamma_k(\pi) - \varepsilon\gamma_k(\pi) \leq 0, \qquad \forall\, p \in P_k(\varepsilon),\ h_p > 0,\ \forall\, k; \qquad (4.2)$$

where $c_p(\pi)$ and $\gamma_k(\pi)$ are the functions which give the cost of path p and the cost of the minimum cost path for commodity k when π varies. It can be very difficult to formulate constraints (4.2) as γ_k is not related to a fixed path.

Consider the particular case where each o/d path p uses portions of at most two trips, namely i and j. Let $g(p,i,j,\pi_i,\pi_j) = (c_p(\pi) - \gamma_k(\pi)) / \gamma_k(\pi) - \varepsilon$. Note that $g(p,i,j,\pi_i,\pi_j)$ can be easily computed when the variations π_i and π_j are applied to i and j. Hence we can impose the following constraints:

$$g(p,i,j,\pi_i,\pi_j) \leq 0, \qquad \forall\, p\colon h_p > 0. \qquad (4.3)$$

Now a model based on Quadratic Semi-Assignment [7] can be introduced. Consider a bipartite graph (S,T,W) where nodes in S correspond to trips and nodes in T correspond to possible trip departure time variations. The forward star of $i \in S$ identifies all possible departure time variations for trip i. This corresponds to a discretization of time window constraints of problem (4.1). Consequently, restrictions illustrated in the first example, where time window discretization depends on the trip, can be approached by appropriately defining W. A semi-assignment is a subset of W with exactly n arcs no two of which being incident on a same node of S. Hence a semi-assignment defines the variation of departure times of all the trips. We will denote with $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ trip departure time variations provided by a feasible semi-assignment $\{(i,\pi_i) \in W,\ i=1,\ldots,n\}$, and with Π the set of all feasible semi-assignment.

Let $P(i,j)$ be the set of paths p using the pair of trips (i,j) such that $h_p>0$, and Δ_p be the

generalized cost modification of p due to π_i and π_j.
The interaction cost of this pair is denoted by $q_{ij\pi_i\pi_j}$ and is defined as follows.

$$q_{ij\pi_i\pi_j} = \begin{cases} \sum_{p\in P(i,j)} \Delta_p\, h_p & \text{if } g(p,i,j,\pi_i,\pi_j)\leq 0,\ \forall\ p\in P(i,j), \\ M & \text{if } \exists\ p\in P(i,j)\text{: } g(p,i,j,\pi_i,\pi_j)>0, \\ 0 & \text{otherwise,} \end{cases}$$

where M is a suitably large penalty value.
Hence the model can be written as follows:

$$\min \sum_{ij} q_{ij\pi_i\pi_j} \qquad (4.4)$$
$$\pi\in\Pi.$$

Consider a pair of arcs (i,π_i) and (j,π_j) in a semi-assignment. If $g(p,i,j,\pi_i,\pi_j) \leq 0$ for all paths in $P(i,j)$ then the variations of cost of each path $p\in P(i,j)$ multiplied by number of passengers is added to the objective function. Otherwise, if $g(p,i,j,\pi_i,\pi_j) > 0$ for at least one $p\in P(i,j)$, the penalty M is added, as constraints are violated. Note that constraints on the vehicle blocks and driver duties can be included in the formulation adding a suitable penalty in the objective function; in particular we can set $q_{ij\pi_i\pi_j}= M$, if $\pi_j-\pi_i<s_{ij}$, $\forall\ (i,j)\in A_3\cup A_4$.
Hence, solving (4.4) provides trip departure times modifications such that:

- the given vehicle blocks and driver duties modified according to π, are still feasible;
- the starting assignment h is an ε-feasible flow;
- the generalized cost of assignment h is minimized;
- the time variations assume values in the given sets.

If we consider the general case, that is when o/d paths can be composed by more than two portions of trips, model (4.4) is only an approximation of the real case. In fact, constraints (4.3) provide only an approximation of (4.2) as it considers only pairs of trips.
The Quadratic Semi-Assignment problem is in general difficult to solve optimally, even though many classes of easy instances have been provided [8]. Branch&Bound algorithms are the most efficient solution methods, but they are able to solve only instances of small size; however, many efficient heuristic procedures have been proposed and, in practice, they produce solutions very close to the optimum.

5. ε-equilibria

In order to characterize suitable values of the tolerance ε, let us now introduce a new definition of assignment, called *minimal ε-equilibrium*.

Definition 3

A tolerance ε is called a *minimal tolerance* if $F(\varepsilon)\neq\emptyset$ and $F(\varepsilon')=\emptyset$, $\forall\ \varepsilon'<\varepsilon$. Any passenger assignment $h\in F(\varepsilon)$ is a *minimal ε-equilibrium flow*.

Note that if h is a minimal ε-equilibrium flow then the set of perceived equivalent paths for the commodity k is given by $P_k(\varepsilon)$. Moreover, $h_p > 0$ implies that $p \in P_k(\varepsilon)$.

Assuming that F in (3.1) is not empty, the procedure in table 1 iteratively increases the value of ε until an ε-feasible flow is found, then returns the minimal tolerance ε. Note that testing $F(\varepsilon)=\emptyset$ can be done by using an LP code. Moreover, since at each iteration the value of ε is increased, new ε-feasible paths are detected and new variables h_p are added to the previous problem. This suggests to use a column generation approach to enhance the performance of the LP code. Finally, to increase the value of the tolerance it suffices to find the minimum path cost for each commodity k, $c(k)$, in the set of all feasible paths having tolerance greater than ε. This can be computed by enumerating the r-shortest paths within that set.

```
Procedure Minimal($\varepsilon$):
begin
    $\varepsilon$:=0;
    while $F(\varepsilon)=\emptyset$ do
        begin
            for $k$:=1 to $K$ do $c(k) := \min\{c_p: p \in P_k \setminus P_k(\varepsilon)\}$;
            $\varepsilon := \min\{(c(k) - \gamma_k)/\gamma_k, k=1,\dots, K\}$
        end
end.
```

table 1

Note that set of all possible relative tolerance values ε is finite and it depends on the number of feasible paths, procedure Minimal(ε) is polynomial in the number of enumerated paths which, in real applications, is usually quite small. Hence the following proposition holds.

Proposition 1

A minimal tolerance ε and a corresponding minimal ε-equilibrium flow can be found by procedure Minimal(ε).

Assume that an exogenous ε has been provided and ε_m denotes the minimal tolerance returned by the procedure Minimal(ε); if $\varepsilon<\varepsilon_m$ then no feasible passenger assignment exists for that given tolerance. If model data (demand, network and costs) are correct, then either the tolerance ε has been underestimated or there exists a Wardrop equilibrium assignment where some passengers use paths whose perceived cost is strictly greater than the cost of minimum equivalent paths, that is $(1+\varepsilon)\gamma_k$. This problem is addressed in [12].

Conversely, if $\varepsilon>\varepsilon_m$ then in any assignment $h \in F(\varepsilon_m)$ passengers do not travel on paths belonging to $F(\varepsilon)\backslash F(\varepsilon_m)$, even if those paths are perceived equivalent; that is, paths with relative cost greater than ε_m are not used. In this case we think that this assignment more precisely reflects the passenger behavior since the tolerance estimation is a difficult task and it can be affected by errors.

In the following, ε will denote the value of a minimal tolerance and $F(\varepsilon)$ will denote the set of all minimal ε-equilibrium flows. Since in general $F(\varepsilon)$ may contain many equivalent assignments, one can try to further characterize one particular ε-equilibrium flow.

We start by considering the passenger behavior assumptions. Since in any assignment in $F(\varepsilon)$ passengers of the same group use paths which are perceived as equivalent, a Wardrop user equilibrium assignment can be used to select an assignment strongly related to the passengers behavior. This can be done by relaxing capacity constraints, that is by introducing a penalty function for each in-vehicle arc. Formally, denoting by v_e the flow on arc e:

$$v_e = \sum_{k=1}^{K} \sum_{p\in P_k(\varepsilon)} h_p\delta_{ep} \le u_e, \qquad \forall\ e\in E_v,$$

let $g_e(v_e)$ be the generalized cost perceived by a passenger travelling on arc e, when the flow is v_e (see fig. 2). Function $g_e(v_e)$ is strictly increasing, with $g_e(0)=c_e$, where c_e is the travel cost without congestion on arc e, and $g_e(v_e)$ grows very fast as soon as v_e approaches the vehicle capacity u_e.

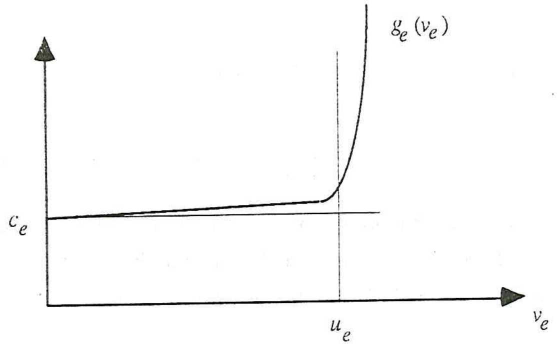


fig. 2

Recalling that E_v,represents the set of all in-vehicle arcs, the Wardrop user equilibrium assignment can be written as follows:

$$\begin{aligned}
&\min \sum_{e\in E_v} \int_0^{v_e} g_e(x)dx + \sum_{e\in E\setminus E_v} c_e v_e \\
&\sum_{p\in P_k(\varepsilon)} h_p = D_k, && k=1,\ldots,K; \\
&v_e = \sum_{k=1}^{K} \sum_{p\in P_k(\varepsilon)} h_p\delta_{ep}, && \forall\ e\in E_v, \qquad\qquad (5.1) \\
&h_p \ge 0, && \forall\ p\in P_k(\varepsilon),\ k=1,\ldots,K.
\end{aligned}$$

Equilibrium assignment algorithms [4, 5, 9] are required to solve problem (5.1). Unfortunately, the solution $\bar{h}$ of (5.1) may violate some of the capacity constraints, hence in principle $\bar{h} \notin F(\varepsilon)$. However, since $F(\varepsilon)\neq\emptyset$, an appropriate selection of $g_e(v_e)$ could, in general, provide a solution which violates capacity constraints only by a negligible amount. Moreover, as $\bar{h}$ is a Wardrop user equilibrium assignment, it reflects passenger behavior better than any ε-feasible flow.

Hence all these properties suggest to use this assignment as input for the improvement model proposed in section 4.

Another possible ε-equilibrium which can be used to apply the improving effectiveness model is obtained by considering the following system optimum problem.

$$\min\{\sum_{k}\ \sum_{p\in P_k(\varepsilon)} (c_p/\gamma_k)\, h_p\colon h\in F(\varepsilon)\}. \tag{5.2}$$

The use of relative cost $c_p\ /\ \gamma_k$ in the objective function is motivated as follows. Assume that absolute costs c_p are used; then the solution of (5.2) will provide an assignment where passengers of commodities with "large" γ_k will be assigned to the shortest paths penalizing passengers of commodities with "small" γ_k. This corresponds to assign a priority to passengers of "distant" o/d pairs which cannot be motivated in terms of passenger behavior. When a relative cost is used, passengers use a path according to the relative cost which is independent from the o/d pair.

The solution of (5.2) is an assignment which minimizes the total cost. Let h be an optimal solution of (5.2) and let r_p be the residual capacity of $p\in P_k(\varepsilon)$:

$$r_p = \min\{u_e - v_e\colon e\in E_v \cap p\}.$$

The following proposition is a direct consequence of optimality of h.

Proposition 2

Given a commodity k, if there exists $p\in P_k(\varepsilon)$ such that $r_p > 0$ then $h_{p'} = 0$, $\forall\ p'\in P_k(\varepsilon)$ such that $c_{p'} > c_p$.

By summarizing, the assignment provided by (5.2) is an ε-feasible flow which minimizes the total cost of the users, and has the extremality property given in proposition 2, that is, for each commodity the cheapest paths are saturated. These properties suggest that also the solution of the system optimum problem can be used as input for the improvement model discussed in section 4.

Finally, the comparison of results obtained using models (5.1) and (5.2) as starting pattern for the conservative model, may suggest useful guidelines in further planning analyses. In fact, using ε-feasible or ε-equilibrium flows one can analyze how passengers react to the service changes in order to minimize their own travel cost. On the other hand, a system optimum assignment allows to evaluate scenarios where one could force passenger choices in order to minimize the total travel cost.

Conclusions

A model to improve the effectiveness of the transit system in a regional area has been proposed. Resource constraints are taken into account in a "conservative" fashion, since only variation of trips departure times which maintain the feasibility of the given vehicles and drivers scheduling are considered. Under quite reasonable hypotheses, this model yields a network flow problem, while, when some hypotheses are relaxed, the model gives a Quadratic Semi-Assignment

problem. The proposed model takes in input a passenger assignment (either ε-feasible or ε-equilibrium) which satisfies the vehicles capacities. An assignment is determined by finding a feasible solution of a multicommodity flow problem. The definition of an objective function helps to characterize the passenger assignment. Here we proposed several approaches.
Further work will be devoted to an experimental evaluation of the approach and to provide a non conservative model to obtain an assignment where the perceived cost for each user is not worse than the starting one, possibly changing the vehicles and the drivers scheduling.

References

[1] Ahuja, R.K., T.L. Magnanti, and J.B. Orlin, "Network Flows. Theory, Algorithms and Applications" (1993), Englewood Cliffs NJ: Prentice Hall.

[2] Alfa, A., "Departure rate and route assignment of commuter traffic during peak period" *Transportation Research*, (1989) **23B**, 337-344.

[3] Carraresi, P. and G. Gallo, "Network models for vehicle and crew scheduling" *EJOR*, (1984) **16**, 139-151.

[4] Florian, M., "An introduction to network models used in transportation planning", in *Transportation Planning Models*, M. Florian, Ed. (1984), North-Holland: Amsterdam, 137-152.

[5] Magnanti, T.L., "Models and algorithms for predicting urban traffic equilibria", in *Transportation Planning Models*, M. Florian, Ed. (1984), North-Holland: Amsterdam, 152-185.

[6] Mahmassani, H.S. and R. Herman, "Dynamic user equilibrium departure time and route choice on idealized traffic arterials" *Transportation Science*, (1984) **18**, 362-385.

[7] Malucelli, F., "A polynomially solvable class of Quadratic Semi-Assignment Problems" (1994), Dipartimento di Informatica - Università di Pisa.

[8] Malucelli, F. and D. Pretolani, "Quadratic Semi-Assignment Problems on structured graphs" *Ricerca Operativa*, (1994) **69**.

[9] Nguyen, S., "A unified approach to equilibrium methods for traffic assignment", in *Traffic equilibrium methods*, M. Florian, Ed. (1976), Springer-Verlag: Berlin, 148-182.

[10] Nguyen, S. and S. Pallottino, "Equilibrium traffic assignment for large scale transit network" *EJOR*, (1988) **37**, 176-186.

[11] Nguyen, S. and S. Pallottino, "Demand estimation and passenger assignment", in *AIRO '90 Annual Conference* (1990) Sorrento (Italy), 391-392.

[12] Nguyen, S., S. Pallottino, and F. Malucelli, "A modelling framework for passenger assignment on transport network with time-tables" (1994), working paper.

[13] Sumi, T., Y. Matsumoto, and Y. Miyaki, "Departure time and route choice of commuters on mass transit systems" *Transportation Research*, (1990) **24B**, 247-262.

PARALLEL SESSIONS

Efficient Heuristics for Traveling Salesman
and
Vehicle Routing Problems with Time Windows

by

Michel Gendreau[1,2] *(speaker)*
Alain Hertz[3]
Gilbert Laporte[1]
Stan Mihnea[4]

[1] Centre de recherche sur les transports, Université de Montréal
[2] Département d'informatique et de recherche opérationnelle, Université de Montréal
[3] Département de mathématiques, École Polytechnique Fédérale de Lausanne
[4] Département de mathématiques appliquées, École Polytechnique de Montréal

1. Introduction

The *Traveling Salesman Problem* (TSP) and the *Vehicle Routing Problem* (VRP) play a central role in distribution planning and have been studied extensively over the past four decades. In recent years, several new heuristics have been proposed to determine good approximate solutions for these difficult problems. Among these, the GENIUS method (for the TSP) and the TABUROUTE algorithm (for the VRP) of Gendreau, Hertz and Laporte [1,2] have displayed interesting computational results when applied to the classical versions of the problems. It is well-known, however, that these classical versions of the TSP and the VRP, while encompassing several of the key dimensions of distribution problems, do not represent adequately many real-life situations since they do not account for some of the critical constraints that severely restrict the design of distribution routes in practical settings. In particular, *time window constraints* that indicate when customers can be visited by vehicles do not appear in the classical versions of the TSP and the VRP. The purpose of this talk is to describe extensions of GENIUS and TABUROUTE that can be used to solve respectively the *Traveling Salesman Problem with Time Windows* (TSPTW) and the *Vehicle Routing Problem with Time Windows* (VRPTW).

This extended abstract is organized as follows. The TSPTW and VRPTW are formally defined in section 2. We then briefly describe GENIUS and TABUROUTE in section 3. The modifications to these heuristics required to handle time windows are discussed extensively in section 4. Section 5 is devoted to the computational testing of the heuristics.

2. The TSPTW and VRPTW

The VRPTW is defined on a graph $G = (V, A)$, where $V = \{v_1, ..., v_n\}$ is a vertex set and $A = \{(v_i, v_j) : i \neq j,\ v_i, v_j \in V\}$ is an arc set. Vertex v_1 is a *depot* at which are based m identical vehicles, where m is either *fixed* or *bounded* above by $\overline{m}$. Vertices $v_2, ..., v_m$ represent *customers* who must be visited by the vehicles. With every arc (v_i, v_j) is associated a non-negative cost c_{ij} representing the travel time from v_i to v_j. With each customer v_i are associated a non-negative demand q_i, a service time δ_i and a time window $[a_i, b_i]$ specifying when the vehicles may visit the customer. The VRPTW consists of designing a set of least cost routes in such a way that a) every route starts and ends at the depot; b) every customer is visited exactly once by exactly one vehicle; c) the total demand carried by any vehicle does not exceed the vehicle capacity Q, and d) given that vehicles depart from the depot at time 0, the arrival times of vehicles at customer locations fall within their time windows (vehicles are allowed to wait if they arrive too early). In addition, one may require the length of any route not to exceed a preset upper bound L. With respect to the objective, the cost of a route is given either by the sum of travel times of the arcs included in it or by its total duration (including any time lost waiting).

The TSPTW can be viewed as a special case of the VRPTW in which there is a single vehicle of unlimited capacity that must visit all customers.

The classical versions of the TSP and the VRP differ from the TSPTW and the VRPTW only by the absence of the time window constraints.

3. GENIUS and TABUROUTE

GENIUS is a two-phase heuristic for the TSP consisting of a tour construction phase based on a generalized insertion step (GENI) followed by a tour improvement or post-optimization procedure (US).

At a general step of GENI, some vertices already belong to a partial tour while others are free. To perform a generalized insertion, consider the partially constructed tour $(v_1, v_2, ..., v_{t-1}, v_{t+1}, ..., v_h, v_1)$ with a given orientation. For any vertex v, define $N_p(v)$ as the set of the p vertices closest to v already on the tour (if $p \leq h$), or as the set of all vertices on the tour (if $p > h$). Let P_{rs} be the set of vertices on the path from v_r to v_s for a given orientation of the tour. For a vertex v not yet on the tour, GENI considers two types of insertion:

Type I: Select vertices $v_i, v_j \in N_p(v)$ and $v_k \in N_p(v_{j+1}) \cap P_{ji}$. Delete arcs $(v_i, v_{i+1}), (v_j, v_{j+1})$ and (v_k, v_{k+1}); insert arcs $(v_i, v), (v, v_j), (v_{i+1}, v_k)$ and (v_{j+1}, v_{k+1}).

Type II: Select vertices $v_i, v_j \in N_p(v), v_k \in N_p(v_{i+1}) \cap P_{ji} \backslash \{v_j, v_{j+1}\}$ and $v_l \in N_p(v_{j+1}) \cap P_{ij} \backslash \{v_i, v_{i+1}\}$. Delete arcs $(v_i, v_{i+1}), (v_j, v_{j+1})$ and (v_{l-1}, v_l) and (v_{k-1}, v_k); insert arcs $(v_i, v), (v, v_j), (v_l, v_{j+1}), (v_{k-1}, v_{l-1})$ and (v_{i+1}, v_k).

To determine the best move, it is necessary to compute the cost of the tour corresponding to each insertion, to each orientation of the tour, and to each possible choice of v_i, v_j, v_k, v_l. GENI can be executed in $O(np^4 + n^2)$ operations. In the post-optimization phase US, each vertex is in turn removed from the tour which is then reoptimized locally, using the reverse GENI operation, and the vertex is then reinserted in the tour using GENI. The procedure ends when it yields no further improvement.

TABUROUTE is a solution improvement heuristic for the VRP based on the *tabu search* approach proposed by Glover [3,4]. It is an iterative search scheme in which a set of neighbours of the current solution are examined at each iteration and the best is selected, even if it leads to a deterioration of the objective. In this way, local optima are avoided, but cycling becomes possible. To prevent such an occurrence, a short-term history of the search trajectory is maintained in *tabu lists* and moves to recently visited solutions are forbidden.

At a general step of TABUROUTE, consider the current solution and randomly select q vertices among a subset of $V \backslash \{v_1\}$. For each selected vertex, compute the cost of the solution obtained by removing it from its current route, and inserting it using GENI in another route containing one of its p closest neighbours. Perform the best non-tabu insertion. Whenever a vertex v is moved from route r to route s, its reinsertion in route r is tabu for the next θ iterations, where θ is randomly selected in some interval $[\theta_{\min}, \theta_{\max}]$. TABUROUTE contains several other features. a) Initially, $\sqrt{n}/2$ trial initial solutions are created and a limited search is conducted for each of them; the most promising solution is then selected as a starting point for the algorithm. b) Route infeasibilities due to excess weight or excess length are allowed during the course of the search; the excess capacity and length are multiplied by two positive penalty factors α and β which are self-adjusted during the course of the algorithm. c) Whenever the best potential move produces a better incumbent, it is implemented even if it is tabu. d) *Diversification* is used: as the search progresses, vertices which have not often been moved are given a greater likelihood of being selected for reinsertion in a different route. e) *Intensification* is also used: when the main search is completed, an intensive search of short duration is carried out in order to improve upon the current best solution. f) When the search is completed, each individual route is reoptimized using the US procedure.

4. Handling Time Windows in GENIUS and TABUROUTE

The introduction of time window constraints in traveling salesman and vehicle routing problems modifies these problems in several ways. It first induces a strong asymmetry in the problem structure, since it now becomes extremely important to trace routes starting from the depot and back to it accounting for the passage of time as one goes from one customer to the next. Several solutions that would be feasible in a standard TSP or VRP are now infeasible (this is especially critical for the TSP). The possibility of having to wait at customer locations also means that the impact of inserting a customer in a route is no longer readily evaluated from the travel costs of the arcs involved in the insertion.

In the context of GENIUS and TABUROUTE that rely heavily on the concept of "closest neighbours" to determine which tentative solutions are examined at any step, a more subtle, yet critical, consequence of the presence of time windows is the distortion of proximity relationships among customers: it may no longer be attractive to visit successively customers located close to one another if these have radically different time windows.

To address these difficulties, the definition of neighbourhoods has been substantially altered. First, two neighbourhoods are now considered for each customer, an inward one and an outward one, to reflect the asymmetry in the problem structure. Second, to avoid wasteful computations, a customer v_j can be included in the outward (resp. inward) neighbourhood of a customer v_i, only if it is feasible (w.r.t. the respective time windows of v_i and v_j) to visit v_j after (resp. before) v_i. Finally, the inclusion of a customer in a neighbourhood depends both on travel distance and similarity of time windows. Two families of neighbourhoods structures have been examined: *composite neighbourhoods* based on a composite distance function which is a weighted average of travel distance and difference of time window midpoints, and *combined neighbourhoods* which are unions of neighbourhoods defined separately w.r.t. to travel distance and difference in time window midpoints. Computational experiments are now conducted to determine which structure yields the best results.

It is not possible to describe in detail all the modifications that had to be made to the basic versions of GENIUS and TABUROUTE to handle time windows. Let us simply sketch the most important of them:

- In GENI, the order in which customers are introduced in the tour is now based on the time windows, instead of being arbitrary.
- Also in GENI, an elaborate procedure has been developed to handle *deadlocks*, i.e. situations where it is impossible to insert any customer in the current partial tour without violating time window constraints. This procedure involves removing customers already on the tour and proceeding with a modified insertion sequence.
- Additional data structures recording the *available slack* (w.r.t. to time windows) at each customer have been introduced to allow for the efficient evaluation of insertions.

5. Computational results

The new version of GENIUS is being extensively tested on routes produced by other heuristics for the VRPTW problems of Solomon [5]. These tests indicate that our approach can find in many instances solutions that are significantly better than the ones previously known. The results of these tests and of others conducted on larger problems will presented and discussed in the talk.

Computational testing of TABUROUTE-TW will soon get under way. Full computational results on Solomon's problems, which are the standard benchmark for the VRPTW, will be reported at the Conference.

References

[1] Gendreau, M., Hertz, A., Laporte, G., "New Insertion and Post-Optimization Procedures for the Traveling Salesman Problem", *Operations Research* 40, 1086-1094, 1992.

[2] Gendreau, M., Hertz, A., Laporte, G., "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science*, forthcoming.

[3] Glover, F., "Tabu Search, Part I", *ORSA Journal on Computing* 1, 190-206, 1989.

[4] Glover, F., "Tabu Search, Part II", *ORSA Journal on Computing* 2, 4-32, 1990.

[5] Solomon, M.M., "On the Worst-Case Performance of Some Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints", *Networks* 16, 161-174, 1986.

Heuristics For The Vehicle Fleet Mix Problem

Ibrahim H. OSMAN [1] and Said SALHI [2]
[1] Institute of Mathematics & Statistics, University of Kent, Canterbury, UK.
E-mail: io@ukc.a.c.uk
&
[2] School of Mathematics & Statistics, University of Birmingham, Birmingham, UK.
E-mail: salhis@vms1.bham.a.c.uk

Abstract:
We briefly review the state of the art algorithms for solving the vehicle routing mix fleet problem. Furthermore, an efficient heuristic based on tabu search philosophy with enhanced data structure and a modified interactive route perturbation procedure are developed. Computational results are reported on a set of 20 test problems from the literature for which 16 new best known solutions are found.

1 INTRODUCTION

The vehicle fleet mix (VFM) problem is an extension of the classical vehicle routing problem (VRP) in which there is a heterogeneous fleet of vehicles. Each vehicle is characterised by its: carrying capacity, maximum travel time, variable running cost and fixed vehicle cost. Each vehicle route originates and terminates at a central depot in order to service a set of customers with given demands. Each customer must be supplied by exactly one vehicle route. The objective is to find a fleet mix of vehicles and a set of routes that minimize the total variable and fixed costs while satisfying the problem constraints.

The VFM problem is computationally complex, since it generalizes the classical VRP, which is known to be NP-hard, Fisher and Rinnooy Kan [1981]. Consequently it is unlikely that polynomial-time algorithms exist to solve to optimality large instances of the problem. The VFM complexity necessitates the development of effective heuristic procedures that are capable of providing high-quality approximate solutions.

There is a few published works on the VFM problem. Mixed integer formulations are presented in Golden *et al.* [1984], Gheysens *et al.* [1984], Ronen [1992] and Salhi and Rand [1993]. Ferland and Michelon [1988] show that an exact method for the VRP with time windows based on a column generation approach can be extended to solve the VFM but no computational results are reported. Golden *et al.* [1984] describe heuristics based on the savings method and the route first-cluster second procedure. Gheysens *et al.* [1986] develop a cluster first-route second heuristic. Desrochers and Verhoog [1991] present a new savings heuristic (MBSA) based on successive route fusions which are selected by solving weighted matching problems. Recently, Salhi and Rand [1993] develop an interactive perturbation procedure consisting of a series of refinement modules. There is a considerable interest in the design of new heuristic methods for routing and scheduling problems in particular and combinatorial optimization problems in general. For a recent survey on routing and scheduling problems, we refer to Osman [1993a] and the bibliography in Osman and Laporte (1994).

The remaining of the paper is organised as follows. The modified RPERT construction heuristic is described is Section 2. The tabu search algorithm with its embedded data structure are discussed in Section 3. Computational comparisons on a set of 20 standard test problems are reported in Section 4. Finally, Section 5 gives the summary and concluding remarks.

2 MODIFIED RPERT PROCEDURE (MRPERT)

Salhi and Rand [1993] propose an interactive procedure denoted by RPERT which starts by solving a VRP using any given vehicle capacity. It constructs an initial solution using the procedure in Salhi and Rand [1987]. RPERT then attempts to improve upon the initial solutions using seven phases. Each phase utilises a different perturbation module which is applied within the routes in order to improve the vehicle utilisation of the whole fleet. Moreover, RPERT performs one single cycle of search applying each perturbation module once before it terminates the search. The best stored solution at the end of the cycle is the RPERT final solution.

Two modifications are introduced to RPERT to form an improved procedure which is denoted by MRPERT. Let define a move to be a transition from one solution to another of its neighbours according to a neighbourhood generation mechanism. In RPERT, a rigid restriction is used in that only feasible moves were allowed to be generated. To alleviate such a restriction and to enlarge the size of neighbourhood, infeasible moves are allowed to be considered in MRPERT. Further, MRPERT is restarted for at least one more cycle using the best solution obtained at the end of the previous cycle. The search is terminated at the end of a cycle where no single improvement over the best solution is made.

3 TABU SEARCH (TSVFM)

Tabu search (TS) is a novel technique for solving hard combinatorial optimization problems. Its goal is to emulate intelligent uses of memory for exploiting structural information. Tabu search ideas was proposed by Glover [1986] and can be viewed as an iterative local search technique. TS explores solutions space by repeatedly making moves from one solution S to another solution S' in the neighbourhood of $S, N(S)$, according to some guiding procedures in order to avoid bad local optima inherited in a local search descent method. We refer to Glover *et al.* [1993] for a user guide on tabu search and to bibliography in Osman and Laporte [1994] for more details on applications.

We sketch below our TS procedure for the VFM.

Tabu Search Procedure (TSVFM).

```
Step 0:   Initialization:
          - Generate an initial solution, $S$, for the VFM problem.
          - Set the best solution $S_{best} = S$.
          - Evaluate all the moves in the neighbourhood $N(S)$.
          - Set values for: the tabu list (TABL); the tabu-list size ($|T_s|$) and the Data Structure (DS) matrices
            for the candidate list of solutions.
          - Set iteration counters: nbiter=0 (current iteration)
            and bestiter= 0 (iteration giving best solution).
Step 1:   Guided Search:
          - Determine strategically, by use of a special data structure DS, the exact set of the candidate
            list of best moves in the neighbourhood, i.e., $N'(S) \subseteq N(S)$.
          - Update DS, if necessary, after each iteration.
Step 2:   Selection Strategy:
          - Choose the best admissible solution $S' \in N'(S)$
            (a non tabu move or tabu but accepted by the aspiration criterion).
          - Set $S = S'$ and nbiter= nbiter +1.
          - If $C(S') < C(S_{best})$, then set $S_{best} = S'$ and bestiter= nbiter.
          - Update the tabu list, TABL.
Step 3:   Stopping criterion:
          If { (nbiter - bestiter) > MAXBEST, a given maximum iteration number}, then stop,
          Else go to Step 1.
```

We shall give a brief description of the basic components of the above TS procedure.

Initial solution:
The initial solution S can be generated using any VFM or VRP heuristic. In this study, we have used the VRP procedure of Salhi and Rand [1987] to generate an initial solution using a given vehicle capacity. The same initial solution is used to start the MRPERT procedure.

Neighbourhood generation mechanism:
Given a VFM solution $S = \{R_1, \ldots, R_p, \ldots, R_v\}$ where R_p is the set of customers supplied by vehicle route p and v is the total number of vehicle routes, we adapt the λ–interchange mechanism with $\lambda = 1$ to generate neighbouring solutions. This mechanism has been defined in its general form in the context of location problems by Osman and Christofides [1994] and successfully implemented different routing and scheduling problems in Osman [1993b, 1993c]; Thangiah, Osman and Sun [1994]; Thangiah, Osman, Vinayagamoorthy and Sun [1994].

Given a pair of route sets R_p and R_q in S, a 1–interchange generation mechanism invokes two processes to generate neighbours. A ***shift process*** reallocates a customer (say i) from one route R_p to another route R_q resulting in two new sets of routes $R'_p = R_p - \{i\}$ and $R'_q = R_q \cup \{i\}$. The ***interchange process*** exchanges two customers say $i \in R_p$ with $j \in R_q$ to get two new route sets $R'_p = (R_p - \{i\}) \cup \{j\}$ and $R'_q = (R_q - \{j\}) \cup \{i\}$. The advantage of mechanism is that the number of vehicles can increase/decrease by one vehicle at a time if necessary in order to find the optimal number of vehicles. The neighbourhood $N(S)$ is defined to be the set of all solutions that can be generated by considering a total number v(v+1)/2 of different pairs of routes involving v non-empty routes and one empty route.

Data structure for the candidate list of moves:
Data structures for strategically defining the set of elite moves were introduced for the VRP in Osman [1993b] and resulted in savings of more than a half of the computation requirement. Here, similar concept is adopted and further enhanced. TS selects the best 1-interchange move, $1IM_{best}$, from S to one of its best neighbours S'. This requires the whole neighbourhood $N(S)$ to be re-evaluated after each performed iteration. Since, the 1-interchange move from S to S' involves only two routes say p and q, the others remain intact. The data structure which can store the set of best candidate moves of the unchanged pairs of routes and strategically updates the set of affected ones is desirable. The data structure uses two levels of memory to avoid unnecessary computation and maintains the exact set of best candidate solutions without relying on random sampling approaches that may miss good solutions.

The first level of memory structure stores the costs of all 1-interchange moves in the neighbourhood of S. For instance, the cost of serving a route, say R_p, without a customer, say i, $C(R'_p)$ can be stored in a cell COST_WITHOUT(p,i) of a v $\times$ n matrix. This value will remain unchanged, once computed at a given iteration, during all other evaluations which shift customer i into other routes (say R_q, $\forall q \in V, q \neq p$). The COST_WITHOUT matrix is also used to evaluate quickly the cost of exchange moves. However, there are few special cases which need to be checked carefully when dealing with exchange moves. The second level of memory structure is mainly to store in a DSTABLE matrix of dimensions $v \times v$ the objective function values of the best v(v+1)/2 moves obtained from the v(v+1)/2 pairs of routes in $N(S)$. After the best move, $1IM_{best}$, is performed and S' becomes the new current solution for the next iteration. S and S' are only different in one pairs of routes, say (R_p,R_q), associated with $1IM_{best}$ and the others remain routes intact. Consequently, 2xv pairs of sets (R_p,R_m) and (R_m,R_q) $(\forall m \in V, m \neq p, m \neq q)$ are necessary to be evaluated in order to update the candidate list of best candidate solutions. Another matrix called DSINF is also used to store other attributes associated with each of the best moves in DSTABLE. For example, these attributes may include the customers exchanged/shifted, the routes involved, insertion positions and others.

Tabu List:
TABL takes the form of an (v+1) $\times$ n matrix (v rows: one for each route set R_p, one for the empty route; n columns: one per customer). In the tabu list, we store some attributes of the accepted 1-interchange move, $1IM_{best}$, from S to S'. These attributes are prevented from being reversed for a certain $|T_s|$ iterations. More precise, TABL(i, p) and TABL(j, q) store the value of nbiter+$|T_s|$ where nbiter is the current iteration number. Customers i and j are allowed to return to R_p and R_q, respectively after nbiter+$|T_s|$ iterations are preformed. A 1-interchange is considered tabu if i is returned to R_p and j is returned to R_q and its status can be checked using a simple test:

$$\text{nbiter} > \text{TABL}(p,i) \text{ and nbiter} > \text{TABL}(q,j)$$

If a move is considered tabu but it leads to a new solution which is better than the best found so far, then its tabu status of the move is dropped since the new direction guarantees no cycling. The tabu list size seems to be related to tabu conditions, selection strategy neighbours and problem characteristics. Based on our experience with such a neighbourhood mechanism and tabu conditions $|T_s|$ can take any values $\left\lceil \frac{n}{p} \right\rceil$ for p varying between 3 and 5.

Stopping Criterion:
The TS procedure is terminated after the number of iterations without any improvement over the best solution so far is greater than a given value of MAXBEST= 5 $\times$ n.

4 COMPUTATIONAL EXPERIENCE

A set of 20 standard test problems proposed in Golden *et al.* [1984] with size varying from 12 to 100 customers are used to test our heuristics. In this study, we report the actual solution, their average relative percentage deviation (ARPD) over the new best solutions and the number of best solutions achieved by a particular heuristic. In Table 1, our heuristic solutions are given. The first and second columns give the problem number and the number of customers, respectively. The results of RPERT are in Salhi and Rand [1993], our results for the modified RPERT (MRPERT) and the tabu search (TSVFM) with their respective CPU computation time in seconds on a VAX 8700 computer are next in the table. The OLB BEST are the results of the best known solutions. These results can be found in either Golden *et al.* [1984], Gheysens *et al.* [1984], Desrochers and Verhoog [1991], or Salhi and Rand [1993]. The final column contains the NEW BEST known solutions adjusted to reflect our findings. The new best solutions are indicated in bold. It can be seen that TSVFM finds 16 new best solutions with an average percentage of 0.18%. MRPERT however improves the solutions of RPERT in all cases and produces 7 new solutions with an ARPD of 0.37% compared to 4 best solutions and an ARPD of 0.88% for RPERT. The other heuristics by Golden *et al.* [1984], Gheysens *et al.* [1984], Desrochers and Verhoog [1991] produced 1, 2 and 3 best solutions with an ARPD of 1.50%, 0.91% and 1.54 respectively.

5 CONCLUSION AND FUTURE DIRECTIONS

We have reviewed the literature for the vehicle mix fleet problem. Two heuristics are introduced and compared against the best algorithms in the literature on a set of 20 test problems. The Tabu Search algorithm with the appropriate data structure has obtained interesting results as new best known solutions are found for 14 test problems. The modified interactive route perturbation procedure has produced more best known solutions than the route perturbation procedure of Salhi and Rand [1993] by enlarging the neighbourhood with infeasible moves and restarting for a few entire cycles until no further improvement is found. It was interesting to notice that the Tabu Search algorithm has failed to find the best known solutions obtained by the second MRPERT procedure for two problem instances with a large number of vehicle types. The results form part of an on-going research on this problem. We hope to be able to investigate the reason of such a failure and if it can be resolved. We are also investigating other hybrid heuristic approaches similar to that in Thangiah, Osman and Sun [1994].

Table 1. Comparison of the best known solutions

No.	Size	RPERT	MRPERT	CPU	TSVFM	CPU	Old Best	New Best
1	12	614	606	0.6	606	7	**602**	602
2	12	**722**	**722**	0.4	**722**	6.9	**722**	**722***
3	20	1003	972	2.3	971	19.6	**965**	965
4	20	6447	6447	1.0	**6444**	30.9	6446	**6444***
5	20	1015	1015	0.6	**1009**	14.1	1013	**1009***
6	20	**6516**	**6516**	0.5	**6516**	22.6	**6516**	**6516***
7	30	7402	7377	15.9	7310	39.0	**7298**	7298
8	30	2367	2352	11.0	**2348**	42.0	2349	**2348***
9	30	**2209**	**2209**	1.1	**2209**	38.1	**2209**	**2209***
10	30	2377	2377	1.3	**2363**	37.7	2368	**2363***
11	30	4819	4787	0.7	**4755**	38.2	4763	**4755***
12	30	**4092**	**4092**	0.4	**4092**	45.9	**4092**	**4092***
13	50	2493	2462	25.6	2471	169	**2437**	2437
14	50	9153	9152	32.1	**9126**	179	9132	**9126***
15	50	2623	**2600**	8.9	2607	196	2621	**2600***
16	50	2765	**2745**	91.4	**2745**	154	2765	**2745***
17	75	1767	1767	19.8	**1760**	585	1767	**1760***
18	75	2439	2439	10.45	**2412**	450	2432	**2412***
19	100	8751	8704	24.6	**8681**	792	8700	**8681***
20	100	4187	**4166**	604.1	4189	876	4187	**4166***
No. of	Best	4	7		14		8	16
ARPD%	deviation	0.88	0.37		0.18			

*: best known solution found in this paper.
CPU: The CPU time in seconds.
MRPERT: The modified RPERT in this paper.
Old Best: Best known solutions in the literature.
New Best: New best known solutions up to date.
RPERT: Interactive procedure of Salhi and Rand [1993].
TSVFM: The tabu search procedure.

6 REFERENCES

Desrochers, M., and T.W., Verhoog, 1991. A new heuristic for the fleet size and mix vehicle routing problem. *Computers and Operations Research* 18, 263-274.

Ferland, J.A., and P., Michelon, 1988. The Vehicle Routing With Multiple Types. *Journal of Operational Research Society,* 39, 577-583.

Gheysens, F.G., B.L., Golden and A., Assad, 1984. A comparison of techniques for solving the fleet size and mix vehicle routing problem. *Operations Research Spektrum* 6, 207-216.

Gheysens, F.G., B.L., Golden and A., Assad, 1986. A new heuristic for determining fleet size and composition. *Mathematical Programming Studies* 26, 233-236.

Glover, F., E., Taillard, D., de.Werra, 1993. A user's guide to tabu search. *Annals of Operations Research,* 41, 3-28.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13, 533-549.

Golden, B.L., A., Assad, L., Levy, and F.G., Gheysens, 1984. The fleet size and mix vehicle routing problem. *Computers and Operations Research* 11, 49-66.

Lenstra, J.K., and A.H.G., Rinnooy Kan, 1981. Complexity of vehicle routing and scheduling problems. *Networks,* 11, 221-227.

Osman, I.H., 1993a. Vehicle Routing and Scheduling: Applications, Algorithms and Developments. *Proceedings of the International Conference On Industrial Logistics,* Rennes, France, July 1993.

Osman, I.H., 1993b. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals Of Operations Research,* 41, 421-451.

Osman, I.H., 1993c. Heuristics for the generalised assignment problem: Simulated Annealing and Tabu Search approaches. Forthcoming in *Operations Research Spektrum* (1994).

Osman, I.H., and G., Laporte, 1994. Modern Heuristics for Combinatorial optimization Problems: An annotated bibliography. Forthcoming in *Annals of Operations Research* issue on "Metaheuristic in Optimization", Laporte G., and I.H. Osman (eds).

Osman, I.H., and N., Christofides, 1994. Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. *International Transactions In Operational Research* 1, No. 3.

Ronen, D., 1992, Allocation Of Trips To Trucks Operating From A single Terminal. *Computers and Operations Research,* 19, 451-

Salhi, S., and G.K., Rand, 1987. Improvements to vehicle routeing heuristics. *Journal of Operational Research Society,* 38, 293-295

Salhi, S., and G.K., Rand, 1993. Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research,* 66, 313-330.

Thangiah, S.R., I.H., Osman, and T., Sun, 1994. Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems With Time Windows. Working Paper UKC/IMS/OR94/4, University of Kent, Canterbury, UK, Submitted to *Annals of Operations Research* on "Transportation" Laporte, G., and M., Gendreau (eds).

Thangiah, S.R., I.H., Osman, R., Vinayagamoorthy and T., Sun 1994. Algorithms for Vehicle Routing Problems With Time Deadlines. *American Journal of Mathematical and Management Sciences* 13, No. 3-4,

Routing Models and Solution Procedures for Regional Less-Than-Truckload Operations

by

Cynthia Barnhart
Daeki Kim

The Pierce Laboratory
Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
cbarnhar@athena.mit.edu
(617)253-3815

April 1994

ABSTRACT

Less-than-Truckload (LTL) carriers are required on a daily basis to solve Intra-Group Line-Haul (IGLH) problems. IGLH problems require the determination of routes to service required pickups and deliveries (i.e., 28-foot trailers) at End-Of-Line (EOL) terminals. The objective is to minimize total costs, given that tractors are able to simultaneously transport two trailers and that all pickups and deliveries must be accomplished. In this paper, an approximate IGLH solution approach is presented.

Given pickup and delivery requirements together with relevant distance data, a matching network is constructed in which nodes correspond to sets of pickups and deliveries and links to routes. A minimum weight non-bipartite matching algorithm is solved over this network and the result is an IGLH solution. This solution is improved by again applying a minimum weight matching algorithm, this time to a matching network in which nodes correspond to routes and links to improved routes. Finally, the routes are sequenced so as to achieve balance at each EOL terminal (i.e., empty trailers must be delivered or picked up as necessary to ensure that each EOL terminal has the same number of pickups and deliveries) and to minimize the inventory of empty trailers. The new IGLH solution procedure is tested on randomly generated data and on data provided by a large LTL carrier. Computational tests show that near-optimal solutions are generated rapidly.

1. INTRODUCTION

In typical Less-than-truckload (LTL) operations, where motor carriers haul freight from many origins to many destinations, service is provided using multiple transportation moves or legs. The first and last legs involve transporting shipments a short distance between their origins/destinations and *end-of-line* (EOL) terminals. Each origin-destination pair of EOL terminals, (called a market), has an associated demand specifying the shipments that the carrier must haul from the origin terminal to the destination terminal. Because the shipment demand for an individual market is typically much smaller than the capacity of a trailer, it is economical to combine the demand for several markets at an origin terminal. The consolidated demand is then transported (i.e., the second transportation leg) to an intermediate terminal, referred to as a consolidation center (*CC*). At the *CC*, consolidated shipments on incoming trucks are unloaded, sorted, and each shipment is reloaded onto an outgoing truck bound for the *CC* nearest its destination. (This third transportation leg is referred to as *main-line* transportation.) At the destination *CC*, each shipment is unloaded, sorted, repacked, and transported to an EOL terminal near its destination before the final transportation leg to its destination.

Regulatory changes in recent years, such as relaxed restrictions on the use of twin-trailer trucks (i.e., one tractor pulling two 28-foot trailers, called doubles) or triples (i.e., one tractor pulling three 28-foot trailers), have provided opportunities for LTL carriers to become more efficient. In main-line operations, for example, it is documented in Sheffi and Powell (1985) that the relaxed regulations on 28-foot trailers (or pups) allow desired service levels to be achieved with reduced cost. Pups provide extra carrying capacity per driver and allow flexibility since a tractor may travel alone (bobtail) or may pull one, two, or in some cases, even three pups.

Although the use of pups in local city pick-up and delivery operations is impractical, pups have the potential for reducing *intra-group line-haul* costs, that is costs associated with transportation between EOL terminals and regional *CC*'s. (The set of EOL terminals served by one *CC* constitutes a *group*.) In contrast to main-line operations, however, the magnitude of these savings is unknown. It is our objective in this paper, therefore, to describe models and solution techniques that determine optimal or near-optimal solutions for the intra-group line-haul problem using pups. In Section 1, the intra-group line-haul problem is defined, a problem formulation is presented and relevant literature is reviewed. An approximate model and solution procedure is presented in Section 2. Computational results achieved using randomly generated and *real-world* data are presented in Section 3. Issues concerning model flexibility and adaptability are discussed in Section 4, and finally, conclusions and future research are presented in Section 5.

1.1 The Intra-Group Line-Haul Problem

The intra-group line-haul problem is characterized by a single CC and the EOL terminals it serves. Associated with each EOL terminal, is the number of pups to be picked-up and transported to the CC (called pickups) and the number of pups to be delivered from the CC (called deliveries). If, for some EOL terminal i, the number of pickups exceeds the number of deliveries, say by e pups, then to achieve *balance*, e empty pups must be delivered from the CC or from another terminal to i. Similarly, if the number of deliveries at i exceeds the number of pickups by e, e empty pups must be picked up at i and delivered to the CC or to another terminal. The objective is to determine tractor routes and pup assignments to the tractor routes such that: 1) total costs are minimized; 2) each route begins and ends at the same location; 3) each pickup and delivery is accomplished; 4) tractor capacity (i.e., the number of pups that can be pulled) is never exceeded; 5) trailers are balanced; 6) pickup and delivery time windows are satisfied; 7) level of service requirements are achieved; and 8) driver work restrictions are not violated. We begin by considering a *simplified* version of the intra-group line-haul problem, referred to as the *core problem*, that satisfies only requirements 1, 2, 3, 4 and 5. The enhanced problem including the omitted requirements is discussed later.

Total costs can be expressed in a number of ways. In cases when the equipment is owned by the LTL company, total cost may equal the sum of fixed costs per vehicle plus variable costs per mile. Alternatively, when the transportation service is contracted out, total cost may be expressed only as a cost per mile. We model the intra-group line-haul problem to allow for these (and other) cost structures, requiring only that total cost can be expressed per route.

Before presenting the core intra-group line-haul problem formulation, we introduce the following:

Notations :

G: = (N,L): complete directed network G with node set N and arc set L, where each node corresponds to an EOL terminal or a CC;

A: node-arc incidence matrix for G;

p : vector of number of trailers to be picked up from each EOL terminal;

d : vector of number of trailers to be delivered to each EOL terminal;

c : vector of costs incurred by driving a tractor between two nodes of G; and

k : maximum number of trailers that can be pulled simultaneously by a tractor.

Decision Variables :

t : vector of the number of tractors traveling on each arc in G;

x : vector of the number of *outbound* trailers, i.e., trailers loaded with freight to be delivered to an EOL terminal, traveling on each arc in G;

y : vector of the number of *inbound* trailers, i.e., trailers loaded with freight to be delivered to the CC, traveling on each arc in G; and

e : vector of the number of *empty* trailers (needed for balance) traveling on each arc in G.

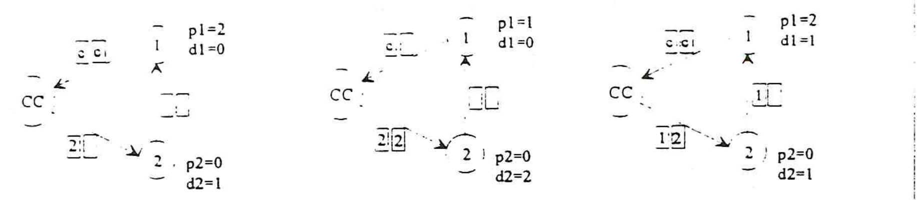
The core intra-group line-haul problem (IGLH) is then formulated as an integer program as follows:

$$\begin{aligned}
&Min \quad c^T t \\
&\text{subject to} \\
&\quad At = 0 && (1)\\
&\quad Ax = -d && (2)\\
&\quad Ay = p && (3)\\
&\quad Ae = d - p && (4)\\
&\quad kt - x - y - e \geq 0 && (5)\\
&\quad t, x, y, e \geq 0 && (6)\\
&\quad t, x, y, e \text{ integer} && (7)
\end{aligned}$$

Constraints (1), (2) and (3) ensure that tractors both enter and leave each location visited and that all trailer deliveries and pickups are accomplished. We assume that the number of trailers to be serviced is known with certainty. This assumption is valid when the modeled is used: a) in a strategic planning mode to develop routes and schedules; and b) in a real-time setting in which new routes and schedules are generated concurrently with the receipt of new pickup and delivery information.

Constraints (4) require that empty trailers are delivered to or picked up from locations in order to achieve balance. Constraints (5) enforce tractor capacity limitations. Except where otherwise noted, we assume that $k = 2$, i.e., that two is the maximum number of trailers that a tractor can carry at any one time. Constraints (6) and (7) ensure that the decision variables are nonnegative and integer, respectively. The objective is to minimize total costs. Distances, often a factor in the determination of total costs, are assumed: 1) to be nonnegative; 2) to be symmetric, that is, the distance from node i to node j is the same as that from j to i; and 3) to satisfy the triangle inequality, that is, the distance from i to k does not exceed the sum of the distances from i to j and from j to k.

To achieve balance in the group, the total number of pickups must equal the total number of deliveries. Thus, it is assumed for the group that if total pickups exceed total deliveries by e, e empty trailers are available at the CC for delivery to EOL terminals and similarly, if total deliveries exceed total pickups by e, it is assumed that e empty trailers are required to be delivered to the depot. To illustrate the notion of empty trailer balancing, consider the example in Figure 1. Nodes CC, 1, and 2 represent the consolidation center and EOL terminals 1 and 2, respectively. Network arcs correspond to tractor movements and the boxes on each arc represent trailers assigned to that tractor movement. A labeled box is a loaded trailer destined for the terminal indicated by the label, while an unlabeled box is an empty trailer.



(a) Total pickups > Total deliveries (b) Total pickups < Total deliveries (c) Total pickups = Total deliveries

Figure 1. Empty Trailer Balancing

In Case (a) of Figure 1 where total pickups (i.e., 2) exceed total deliveries (i.e., 1), empty trailer balancing is achieved by transporting one empty trailer from terminal 2 to terminal 1 and another from the CC through terminal 2 to terminal 1. In Case (b) where total deliveries exceed total pickups, one of the two empties to be picked up at terminal 2 is delivered to terminal 1 and the other is delivered to the CC. Finally, in Case (c) where the total number of pickups and deliveries is equal, empty trailers are neither picked up from nor delivered to the CC. Empty trailer balancing is achieved instead by transporting one empty trailer from terminal 2 to terminal 1.

Without the tractor capacity constraints (5), IGLH decomposes into four, easy network flow problems - one each for the tractors, the pickups, the deliveries, and the empties. With these capacity constraints, however, the pure network structure of IGLH is destroyed and integer programming techniques must be employed to guarantee that an optimal solution is determined. Substantial literature exists describing solution techniques for integer programs, and for vehicle routing and scheduling problems in particular (see for example the comprehensive surveys by Golden and Assad, 1988 and Magnanti, 1981.) For the intra-group line-haul problem in particular, Eckstein(1986) and Eckstein and Sheffi(1987) used optimization techniques, specifically Lagrangian relaxation and branch-and-bound. Their procedure worked well for small problems but was impractical (due to extensive memory requirements and running time) for problems of the size encountered by most LTL carriers. Other than that of Eckstein and Sheffi, we are not aware of any other published accounts of work on the intra-group line-haul problem.

Based on a combination of factors, namely the disappointing results obtained by Eckstein and Sheffi in finding optimal IGLH solutions and the potential need to determine IGLH solutions in real-time, we focused on the development of an approximate IGLH solution technique, described in the next Section.

2. THE INTRA-GROUP LINE-HAUL SOLUTION PROCEDURE

Our approximate solution procedure for the intra-group line-haul problem, denoted IGLH_MATCH, consists of the following three major steps:

STEP 1: IGLH_MATCH Preprocess;

STEP 2: IGLH_MATCH Construct Routes; and

STEP 3: IGLH_MATCH Sequence Routes.

We define a *route* as the sequence of stops visited by one tractor, beginning and ending at a *CC* with one or more intermediate stops at EOL terminals. At each stop, a tractor picks up and/or delivers one or more trailers. A route containing only one EOL terminal is called a *direct*, while one containing more that one EOL terminal is called a *via*. A *loaded direct* is a direct where the tractor carries two deliveries on the route's outbound leg and two pickups on its inbound leg. (The outbound leg is from a *CC* to an EOL terminal, while an inbound leg is the reverse.) A *h-via* is a via visiting *h* intermediate EOL terminals on a route. In the first step of IGLH_MATCH, loaded directs are constructed and the pickups and deliveries assigned to these tractor routes are effectively removed from further consideration. Next, directs and vias servicing every remaining pickup and delivery are constructed. The directs and vias constructed in steps 1 and 2 service each pickup and delivery exactly once. In the final step of IGLH_MATCH, directs and vias are merged together in order to satisfy trailer balancing requirements. We show that balancing can be accomplished without increasing the number of tractor miles traveled. In the following sections, each step of the IGLH_MATCH procedure is detailed.

2.1 Step 1: IGLH_MATCH Preprocess

For each EOL terminal i = 1,2,...,T, let p_i and d_i represent the number of pickups and deliveries respectively, and arbitrarily designate the trailers to be picked up (delivered) as trailer numbers $P_1, P_2, \ldots, P_{p_i}$ $(D_1, D_2, \ldots, D_{d_i})$. Then, the steps of Preprocess are as follows:

Preprocess Step 1: For EOL terminal i = 1,2,...,T, let m_i represent the maximum number of loaded directs between the *CC* and terminal i, that is:

$$m_i = \mathrm{MIN}(\lfloor p_i/2 \rfloor, \lfloor d_i/2 \rfloor);$$

Preprocess Step 2: For tractor $a = 1,2,\ldots, m_i$, assign tractor a to perform a loaded direct transporting trailers D_{2a-1} and D_{2a} from the *CC* to terminal i and transporting trailers P_{2a-1} and P_{2a} from terminal i to the *CC*; and

Preprocess Step 3: Eliminate from the problem data all pickups and deliveries assigned to the loaded directs constructed in Preprocess Step 2. That is, for EOL terminal $i = 1,2,\ldots,T$, let $p_i = p_i - 2m_i$ and $d_i = d_i - 2m_i$.

Thus, Preprocess builds for each EOL terminal, the maximum possible number of loaded directs. The motivation for doing this is derived from the following observation:

Lemma 1:

If $p_i = d_i = 2m_i$ for EOL terminal $i = 1,2,\ldots,T$ and total costs c are proportional to 1) the total tractor miles traveled; or 2) the sum of the total tractor miles traveled plus the fixed costs per tractor, then an optimal IGLH solution is to perform m_i loaded directs for each EOL terminal i.

Proof:

Since the capacity of each tractor is assumed to be two, the minimum number of tractor trips from the *CC* to EOL terminal i is m_i and similarly, the minimum number of tractor trips from terminal i to the *CC* is m_i. Furthermore, since distances are assumed to satisfy the triangle inequality, the minimum distance trip between the *CC* and terminal i is $c_{cc,i}$, i.e., the direct route between *CC* and EOL terminal i. Hence, the optimal IGLH objective function value is at least $\sum_{i=1}^{T} c_{cc,i} * 2m_i$. ∎

To illustrate the preprocessing step, consider Example I in Table 1. EOL terminal 1 requires two pickups and three deliveries. Preprocessing creates one loaded direct to terminal 1, thereby satisfying required pickups and reducing to one the number of remaining deliveries to that location. Preprocessing similarly alters the pickups and deliveries at EOL terminals 2, 3 and 4, as shown in Table 1.

Table 1. Preprocessing of Example I

EOL Number	Number of Pickups Before Preprocess	Number of Deliveries Before Preprocess	Number of Pickups After Preprocess	Number of Deliveries After Preprocess
1	2	3	0	1
2	3	3	1	1
3	3	1	3	1
4	2	2	0	0

Note that after preprocessing, either the number of pickups or the number of deliveries is reduced to 0 or 1.

2.2 Step 2: IGLH_MATCH Construct Routes

After preprocessing, it is necessary in the route construction phase (Step 2) to build additional tractor routes and assign the *remaining* pickups and deliveries to these routes. Like preprocessing, the motivation of the route construction step is to serve the pickups and deliveries at an EOL terminal with the minimum number of tractors. Consider for example, the problem depicted in Figure 2a in which costs are assumed either to be equal to: 1) the total number of tractor miles (in dollars) plus a fixed cost per tractor; or 2) the total number of tractor miles (in dollars) only. Since the total number of pickups equals four and the total number of deliveries is three, at least two tractors are required. The minimum cost solution, depicted in Figure 2a, uses exactly two tractors: one performing the 2-via visiting EOL terminals 1 and 3, and the other performing the direct to EOL terminal 2. Observe that the minimum number of tractors required to service each EOL terminal is one and that exactly this minimum number of tractors visits each EOL terminal.

Consider all alternate solutions (presented in Figures 2b - 2e). In Figure 2b, two tractors visit EOL terminal 2, effectively replacing the optimal tractor movement from terminal 1 to terminal 3 in Figure 2a with the more costly (due to the triangle inequality) two tractor movements, one from terminal 1 to terminal 2 and the other from terminal 2 to terminal 3. Similarly, in Figure 2c, both terminals 1 and 3 are visited by two tractors, thereby altering the optimal solution by replacing the direct to terminal 2 with the more costly (again due to the triangle inequality) 3-via visiting terminals 1, 2 and 3. The solution in Figure 2d is a further degradation of the already nonoptimal solution of Figure 2b since it inserts another visit to terminal 1 along the movement from the *CC* to terminal 2. Finally, Figure 2e represents a solution using 3 tractors, each performing a direct to one of the three terminals. In addition to increasing the number of tractors used, this solution effectively replaces the optimal tractor movement from terminal 1 to terminal 3 with two more costly (due to the triangle inequality) tractor movements, one from terminal 1 to the *CC* and the other from terminal 3 to the *CC*.

Many such examples can be constructed demonstrating that due to the triangle inequality, it is optimal for a tractor visiting an EOL terminal to pickup and deliver a maximal number of trailers. That is, the number of trailers picked up at (delivered to) a terminal by one tractor should equal either the number of pickups (deliveries) required at that terminal or the capacity of the tractor. Although numerous examples can be constructed validating this observation, it is worth noting that there are instances where it is not optimal for each tractor to pickup and deliver a maximal number of trailers at every EOL terminal, i.e., this policy may result in an increase in the minimum number of tractors required.

The observation that it may be optimal for a tractor visiting an EOL terminal to pickup and deliver a maximal number of trailers motivated the use of a matching-based heuristic in the route construction step of the IGLH_MATCH procedure. The matching-based procedure is described in the following sections.

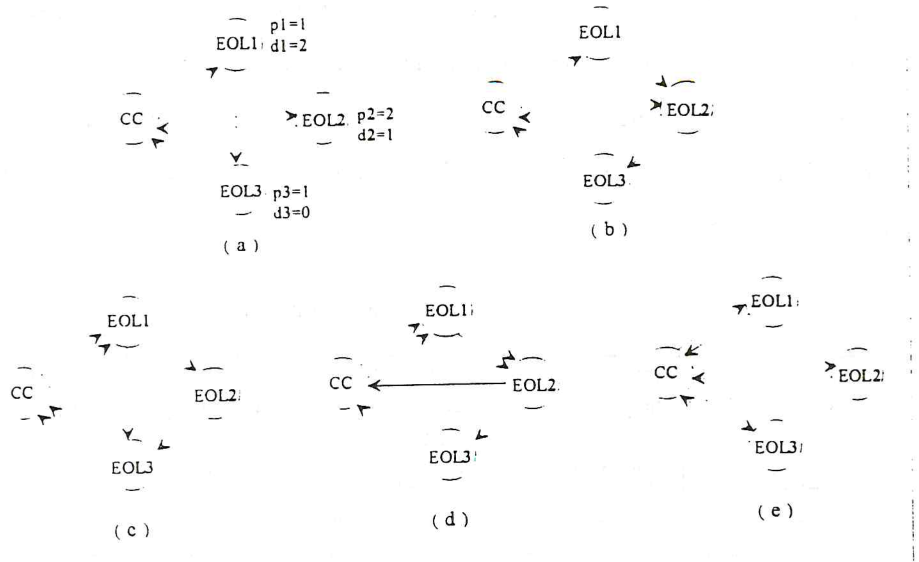


Figure 2. Examples of Alternative Solutions

2.2.1 Solution Procedure

In Construct Routes (Step 2 of IGLH_MATCH), additional tractor routes are built and the *remaining* pickups and deliveries are assigned to these routes. This is accomplished with two procedures: *Match* and *Re-Match*. An initial IGLH solution is determined in *Match*, while *Re-Match* is used to generate an improved solution. We illustrate the *Match* and *Re-Match* procedures using Example I and follow-up with detailed descriptions.

The *Match* Procedure

In the first step of *Match*, the remaining pickups and deliveries (p_i, d_i) at EOL terminal i = 1,2,...,T, are divided into *maximal sets*, that is, into sets where the number of trailers picked up at (delivered to) a terminal by one tractor equals the minimum of: 1) the number of pickups (deliveries) required at that terminal; and 2) tractor capacity. For example, if the remaining number of pickups at terminal 3 is three and the remaining number of deliveries is one, i.e., $(p_3, d_3) = (3, 1)$, then the resulting maximal sets are $(p_3^1, d_3^1) = (2, 1)$ and $(p_3^2, d_3^2) = (1, 0)$. Table 2 shows the maximal sets for Example I (introduced in Table 1).

Table 2. Maximal Sets of Example I

EOL Number	Number of Pickups After Preprocess	Number of Deliveries After Preprocess	Number of Pickups in Maximal Set	Number of Deliveries in Maximal Set
1	0	1	0	1
2	1	1	1	1
3^1	3	1	2	1
3^2			1	0

In the next step, pairs of maximal sets are evaluated to determine if there exists a tractor route that can *legally* service both sets in the pair. To be legal, the route must satisfy tractor capacity limitations. (Additional constraints may also define legality, e.g. driver work and rest rules mandated by government and contractual agreements, time windows restricting the scheduling of service and enforcing minimum service level requirements, etc.) Consider, for example, the pair of maximal sets (2, 1) and (1, 1). This pair is *illegal* since tractor capacity is violated by the required pickup of 3 trailers on the single route.

Every pair is assigned a *weight* equal to the minimum total cost of a legal route associated with that pair. (If the pair is illegal, the assigned weight is set to infinity.) Assuming for now that legality is defined solely by tractor capacity restrictions, Table 3 reports the weights assigned to each pair for Example I.

Table 3. Weights Assigned in the Match Procedure

from \ to	*CC*	EOL 1	EOL 2	EOL 3^1	EOL 3^2
CC	∞	74	100	16	16
EOL 1	74	∞	102	75	75
EOL 2	100	102	∞	∞	44
EOL 3^1	16	75	∞	∞	∞
EOL 3^2	16	75	102	∞	∞

Given the maximal sets and assigned weight for each pair, an *undirected matching* network is constructed: each node corresponds to a maximal set, links are pairs, and each link cost equals the weight of the pair represented by the link. This *base* network is augmented by adding for each base node n, one *copy* node n' and one link from n' to node n. Like the other network links, these additional links correspond to tractor routes. In particular, the link from n' to n represents a direct serving only the trailers at n and the cost of the link is the cost of the corresponding direct. Finally, the network is made complete by adding zero-cost links between every pair of copy nodes and arcs with infinite cost between each pair of nodes with no adjoining link. The Example I matching network is presented in Figure 3 (with some infinite cost links omitted for clarity).

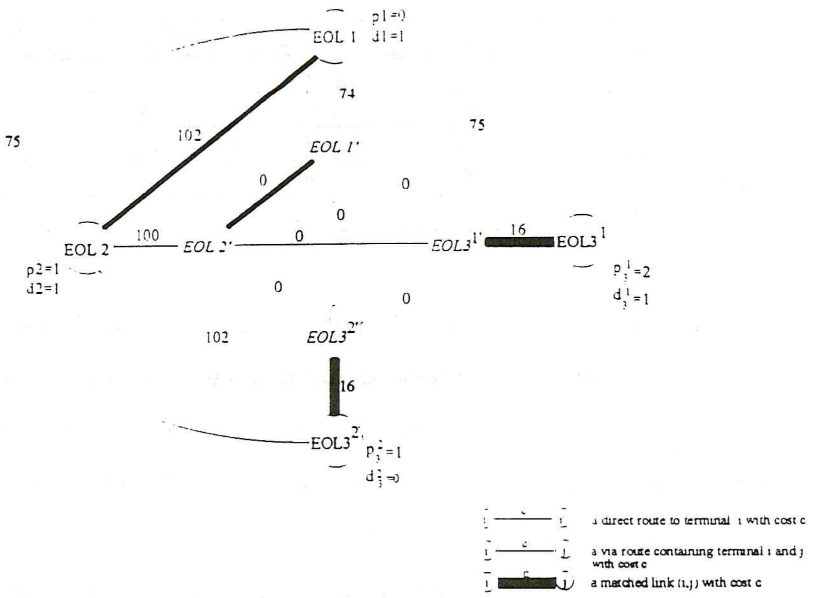

p1=0
EOL 1
d1=1
74
75
102
75
EOL 1'
0
0
0
EOL 2
100
EOL 2'
0
EOL3 1'
16
EOL3 1
p2=1
d2=1
p 1 3=2
d 1 3=1
0
0
102
EOL3 2'
16
EOL3 2
p 2 3=1
d 2 3=0
a direct route to terminal i with cost c
a via route containing terminal i and j with cost c
a matched link (i,j) with cost c

Figure 3. Example I Match Network and Solution

Next, a non-bipartite minimum weight matching algorithm (Gabow, 1973) is used to determine an IGLH solution. By definition, a *matching* in a graph is a set of edges, where no two edges in the matching share the same node. Given weights for the edges, the *minimum weight matching problem* is to find the matching with the smallest sum of weights.

Since the matching network is complete and it contains an even number of nodes (i.e., a copy node is added for each base network node), the matching solution is *perfect*, that is, every node is incident to exactly one arc in the matching. This implies that each pickup and delivery is included in exactly one route. Furthermore, if the weight on each matched link is less than infinity, each pickup and delivery is included in exactly one *legal* route. Since weights on the matching network links equal the costs of the corresponding tractor routes, the optimal matching solution minimizes total costs to service all pickups and deliveries when it is required that: 1) each tractor route contain two or fewer EOL terminals; and 2) each terminal be served by its minimum required number of tractors, or said another way, each tractor serves the maximum possible number of trailers at every terminal.

To illustrate, consider the minimum weight solution in Figure 3 for the Example I matching network. The matched links represent tractor routes that are to be performed. Hence, two tractors perform directs to EOL terminal 3, with one tractor picking up two trailers and delivering one, and the other tractor picking up one trailer. The only other tractor performs a 2-via, first delivering and picking up one trailer each at terminal 2, and then delivering one trailer to terminal 1. The zero-weight matched link connecting copy nodes *EOL 1'* and *EOL 2'* does not correspond to a tractor route, but rather its purpose is to achieve a perfect matching solution. The total cost is 134, the value of the minimum weight matching.

The *Match* procedure is summarized as follows:

***Match* Step 1:** For EOL terminal i = 1,2,...,T, let p_i (d_i) represent the number of pickups (deliveries) remaining at terminal i after preprocessing and let m_i represent the minimum number of tractors needed to service these required pickups and deliveries, that is:

$$m_i = \mathrm{MAX}(\lceil p_i/2 \rceil, \lceil d_i/2 \rceil);$$

***Match* Step 2:** For EOL terminal i= 1,2,...,T, partition the total number of pickups and deliveries (p_i, d_i) into m_i maximal trailer sets, denoted $\{(p_i^1, d_i^1), (p_i^2, d_i^2), \ldots, (p_i^{m_i}, d_i^{m_i})\}$, as follows:

$(p_i^k, d_i^k) = \{\mathrm{MAX}[0, \mathrm{MIN}(2, p_i^0-2(k-1))], \mathrm{MAX}[0, \mathrm{MIN}(2, d_i^0-2(k-1))]\},$

for k=1, 2,...,m_i , where $p_i^0 = p_i$ and $d_i^0 = d_i$.

***Match* Step 3:** Construct the undirected *matching* network as follows: Add one *base* node *n* and one *copy* node *n'* for each maximal set (p_i^k, d_i^k), for k=1, 2, ..., m_i and i= 1,2,...,T; for each base node *n*, add a link between *n* and *n'* with weight equal the cost of the direct between the *CC* and the associated EOL terminal; add a link between each pair of base nodes with weight equal to the minimum total cost legal route associated with that pair; add zero-weight links between every pair of copy nodes; and add infinite-weight links between every pair of nodes without an adjoining link.

***Match* Step 4:** Find the minimum weight matching on the matching network.

The *Re-Match* Procedure

The *Match* procedure limits the number of EOL terminals visited in any route to at most two. Although non-optimal solutions may result, if tractor capacity is the only constraint defining route legality, this limit is not particularly restrictive since 76% of all legal IGLH routes contain two or fewer stops (Table 4). Even so, to reduce the occurrence of non-optimal solutions, a second matching-based procedure, called *Re-Match*, identifies routes that contain up to four EOL terminals and improve the IGLH solution. This improvement is achieved by merging any two routes generated in *Match* into one route *if* the merger is legal and the merged route results in reduced total costs. (It is not necessary to consider routes visiting more than four EOL terminals because there is at least one pickup or delivery at each stop and a tractor can haul at most two pickups and two deliveries in a single route.)

Table 4. All Possible Legal Route Combinations

Number of Stops	Possible Pickups / Deliveries at				No. of Possible Routes
	EOL Terminal 1	EOL Terminal 2	EOL Terminal 3	EOL Terminal 4	
k = 1	(0,1) (0,2) (1,0) (1,1) (1,2) (2,0) (2,1) (2,2)				8
k = 2	(0,1)	(0,1) (1,0) (1,1) (2,0) (2,1)			11
	(0,2)	(1,0) (2,0)			
	(1,0)	(1,0) (1,1) (1,2)			
	(1,1)	(1,1)			
k = 3	(0,1)	(0,1)	(1,0) (2,0)		5
		(1,0)	(1,0) (1,1)		
	(0,2)	(1,0)	(1,0)		
k = 4	(0,1)	(0,1)	(1,0)	(1,0)	1

To illustrate the idea of merging routes, consider the Example I solution generated by *Match*. The 2-via containing EOL terminals 1 and 2, denoted 2V1, and the direct to EOL terminal 3, denoted D1, have a total weight of 118. The pickups and deliveries serviced by these two routes, however, can be serviced with a single *legal* 3-via (denoted 3V1) visiting EOL terminals 2, 1 and 3 (in that order) with total weight of 103, a reduction of about 13%.

The *Re-Match* solution procedure identifies such improvements to the *Match* solution. It does this by first constructing a *Re-Match* network and then solving a minimum weight non-bipartite matching algorithm on this network. Given the *Match* solution consisting of tractor routes and associated pickups and deliveries, an *undirected re-matching* network is constructed: each node corresponds to a tractor route and trailer assignment in the *Match* solution, links are (merged) tractor routes and corresponding trailer assignments, and each link cost equals the minimum cost legal route associated with the (merged) tractor route. For example, routes 2V1 and D1 in the Example I *Match* solution are each represented in the re-match network by a node, and route 3V1 is represented by a link between nodes 2V1 and D1 with weight equal to 103.

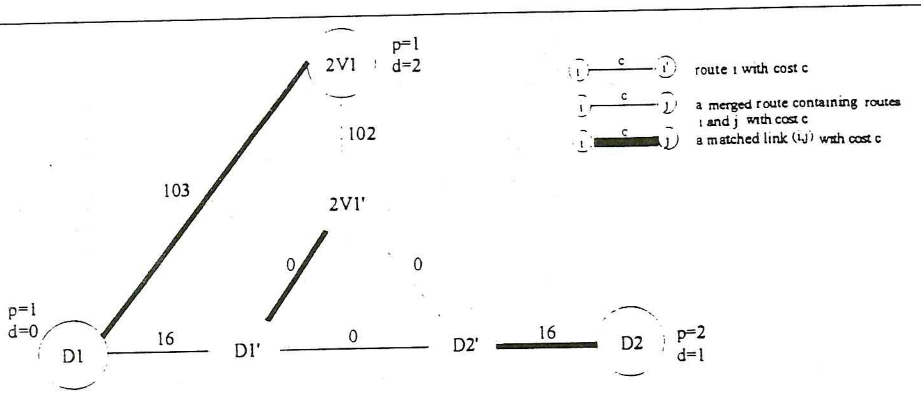

2V1
p=1
d=2
102
2V1'
103
0
0
p=1
d=0
D1
16
D1'
0
D2'
16
D2
p=2
d=1
c
i
i'
route i with cost c
c
i
j
a merged route containing routes i and j with cost c
c
i
j
a matched link (i,j) with cost c

Figure 4. Example I Re-Match Network and Solution

This *base* re-matching network is augmented by adding for each *base* node *r*, one *copy* node *r'* and one link from *r'* to *r*, with cost equal to the weight of the route associated with node *r*. A link from node *r'* to *r* represents exactly the route associated with *r*. Finally, the network is completed by adding zero-cost links between every pair of copy nodes and infinite-cost links between every pair of nodes without an adjoining link.

The Example I re-matching network is presented in Figure 4 (with some infinite cost links omitted for clarity).

Again, since the re-matching network contains an even number of nodes, the *Re-Match* solution is perfect. As in the *Match* solution, this implies that each pickup and delivery is included in exactly one route and, if the weight on each matched link is less than infinity, each pickup and delivery is included in exactly one *legal* route. Since each route *r* in the *Match* solution is represented by a link and two end nodes, and since the weight of that link is equal to the cost of route *r*, the optimal *Re-Match* solution corresponds to tractor routes and associated pickups and deliveries with total cost not greater than that of the *Match* solution. Hence, the *Re-Match* procedure identifies and creates 3- and 4-vias that lead to improved IGLH solutions.

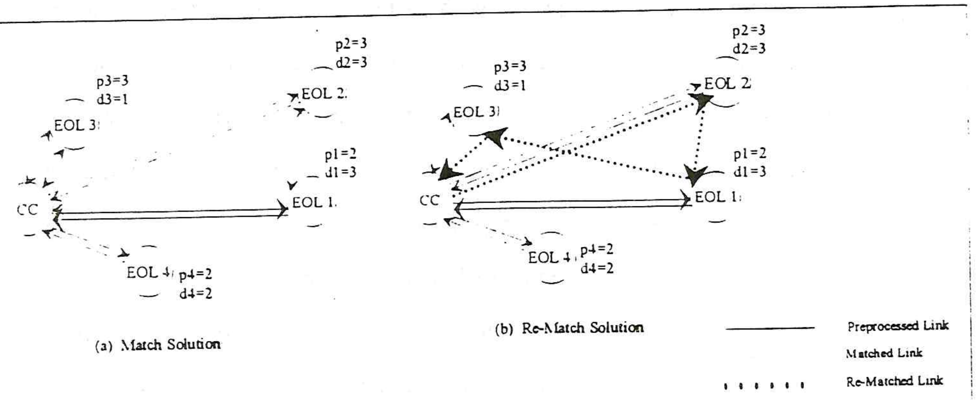
The optimal minimum weight *Re-Match* solution for Example I is depicted in Figure 4. The matched link between nodes *2V1* and *D1*, i.e., the link corresponding to route *3V1*, indicates that the Example I *Match* solution is improved by replacing routes *2V1* and *D1* with route *3V1*. Similarly, the matched link between node *D2* (i.e., the node representing the direct route, denoted *D2*, to EOL terminal 3 with 1 delivery and two pickups) and its copy node *D2'*, indicates that the *Match* solution is unaltered with respect to route *D2*. Finally the link between the two copy nodes *2V1'* and *D1'* is included in the *Re-match* solution only in order to achieve a perfect matching.

The *Re-Match* procedure is as follows:

***Re-Match* Step 1:** Construct the undirected *re-matching* network as follows: Add one *base* node *r* and one *copy* node *r'* for each route generated in the *Match* procedure; for each base node *r*, add a link between *r* and *r'* with weight equal to the weight of the route associated with *r*; add a link between each pair of base nodes with weight equal to the minimum total cost legal route associated with that pair (let this weight equal infinity if no legal route exists); add zero-weight links between every pair of copy nodes; and add infinite-weight links between every pair of nodes without an adjoining link.

Re-Match Step 2: Find the minimum weight matching on the re-matching network.

Figure 5 illustrates the Example I solutions obtained by *Match* and *Re-Match*.



(a) Match Solution

(b) Re-Match Solution

Figure 5. Example I Match and Re-Match Solutions

2.3 STEP 3: IGLH_MATCH Sequence Routes

Constraints (4) of the IGLH problem formulation require the movement of empty trailers in order to achieve balance at EOL terminals. In this section, we present a route ordering procedure called *Sequence Routes*. Using the routes generated in Step 2 of IGLH_MATCH, the *Sequence Routes* procedure accomplishes *empty trailer balancing without additional tractor miles*. Before detailing the procedure, we first provide some notations and assumptions that will facilitate the proof of this result.

Notations :

R: the set of routes generated in Steps 1 and 2 of IGLH_MATCH;

imbalance of route $r \in \boldsymbol{R}$, i.e., $v^r = (\sum_{i=1}^{N^r} p_i^r - \sum_{i=1}^{N^r} d_i^r)$, where i = 1,...,$N^r$ denote the EOL terminals visited in route r (assume that the number of each terminal in r is its stop number, with 0 and N^{r+1} denoting the *CC*), and p^r_i (d^r_i) denote the number of pickups (deliveries) at terminal i accomplished by route $r \in$R;

$\boldsymbol{R}^{\geq}(\boldsymbol{R}^{<})$: the subset of routes contained in $\boldsymbol{R}$ with $v^r \geq (<) 0$;

X^r: MAX$(\sum_{i=1}^{N^r} p_i^r, \sum_{i=1}^{N^r} d_i^r)$, i.e., the total number of loaded and empty trailers assigned to *each* tractor movement along route $r \in \boldsymbol{R}$;

$x^r_{i,j}$: total number of loaded trailers (assigned in Steps 1 and 2 of IGLH_MATCH) on the location i to location j tractor movement along route $r \in \boldsymbol{R}$; and

$e^r_{i,j}$: total number of empty trailers on the location i to location j tractor movement along route $r \in \boldsymbol{R}$, where $e^r_{i,j} = X^r - x^r_{i,j}$.

Assumptions:

Group Balance Assumption: For each route $r \in \boldsymbol{R}^{\geq}$, assume that there are v^r empty trailers available to route r for pickup at the *CC*; and for each route $r \in \boldsymbol{R}^{<}$, assume that $|v^r|$ empty trailers must be delivered by r to the *CC*.

Deliver First Assumption: If the matched link of a *Match* or *Re-Match* solution can represent two routes, each with the same total weight (since distances are assumed to be symmetrical) but with the order of the stops on one route the reverse of the other, the matched link represents the route with the largest difference between the number of deliveries and pickups at the first stop.

For example, consider constructing a route visiting EOL terminal 1 with zero pickups and one delivery and EOL terminal 2 with one pickup and zero deliveries. It is possible to create two legal 2-vias with the same total weight: the first 2-via, denoted r_1, visits EOL terminal 1 followed by EOL terminal 2, and the second, denoted r_2, reverses this direction and visits terminal 2 before terminal 1. The *Deliver First Assumption* states that route r_1 will be generated by IGLH_MATCH since 1 > -1, i.e., the number of deliveries minus the number of pickups at terminal 1 is greater than that same difference at terminal 2.

The following results can now be stated:

Lemma 2: For each route $r \in \boldsymbol{R}$, the total number of loaded trailers assigned to any tractor leg between stops k and k+1 is not greater than X^r, i.e.,

$$x^r_{\mathrm{k.k+1}} = \sum_{i=1}^{k} p_i^r + \sum_{i=k+1}^{N^r} d_i^r \leq X^r, \quad \forall\, k = 0,1,\ldots,N^r, \quad \forall\, r \in R \tag{8}$$

Proof:

By definition of X^r and design of the route set $\boldsymbol{R}$, $X^r = 1$ or $X^r = 2$ for each tractor leg of route r. Clearly, inequality (8) is true if $X^r = 2$ since x^r satisfies tractor capacity restrictions. Consider then, the case when $X^r = 1$ for all legs in route $r \in \boldsymbol{R}$. Clearly the maximum number of EOL terminals visited is two when $X^r = 1$, since each stop has at least one pickup or delivery. For directs, inequality (8) is satisfied since the maximum number of trailers on a tractor leg equals the maximum of p^r_1 and d^r_1. For 2-vias, only one pickup and demand pattern (i.e., one stop with no pickups and one delivery and the other with one pickup and no deliveries) results in $X^r = 1$. Since the *Deliver First Assumption* ensures that all such 2-vias in the IGLH_MATCH solution first visit the terminal with zero pickups and one delivery, $x^r_{CC,1} = 1$, $x^r_{1,2} = 0$, and $x^r_{2,CC} = 1$. ∎

Lemma 3: For each route $r \in \boldsymbol{R}$, let $e^r = X^r - x^r$ where the values of X^r and x^r are as defined above. Given the *Group Balance and Deliver First Assumptions*, the trailer assignments x^r and e^r for each $r \in \boldsymbol{R}$ satisfy the following:

a) loaded trailer pickup and delivery requirements are satisfied;

b) the total number of loaded and empty trailers assigned each tractor leg is less than tractor capacity;

c) empty trailer conservation of flow requirements are satisfied; and

d) the total number of empty trailers assigned each tractor leg is nonnegative.

Hence, empty balancing can be achieved without additional tractor miles.

<u>Proof:</u>

Point a) follows directly from the fact that the loaded trailer assignments for each route $r \in \boldsymbol{R}$ are determined in Steps 1 and 2 of IGLH_MATCH.

By design, each route $r \in \boldsymbol{R}$ is legal and contains no stops with zero pickups and deliveries. Point b) follows from the fact that the total number of trailers a tractor transports between any links of a route r is set to the $\mathrm{MAX}(\sum_{i=1}^{N^r} p_i^r, \sum_{i=1}^{N^r} d_i^r) \leq \mathrm{MAX}(2,2) = 2$.

From point a), it follows that the flow of loaded trailers x^r on route r satisfies conservation of flow requirements, i.e.:

$$x^r_{i-1,i} + p^r_i - d^r_i = x^r_{i,i+1}, \qquad \forall\ i = 1,2,\ldots, N^r,\ \forall\ r \in \boldsymbol{R}.$$

Using the definition $x^r = X^r - e^r$ and substituting, we obtain:

$$X^r - e^r_{i-1,i} + p^r_i - d^r_i = X^r - e^r_{i,i+1},\ \ \forall\ i = 1,2,\ldots, N^r,\ \forall\ r \in \boldsymbol{R}.$$

And rewriting:

$$e^r_{i-1,i} - p^r_i + d^r_i = e^r_{i,i+1}\ ,\ \ \forall\ i = 1,2,\ldots, N^r,\ \forall\ r \in \boldsymbol{R}.$$

Hence, point c) follows since the total number of empties transported on route r into terminal i (i.e., $e^r_{i-1,i}$) plus the number of empties picked up at (i.e., $- p^r_i + d^r_i \geq 0$) or delivered to (i.e., $- p^r_i + d^r_i < 0$) terminal i equals the number of empties transported out of terminal i (i.e., $e^r_{i,i+1}$).

Finally, point d) follows directly from Lemma 2 and the definition $e^r = X^r - x^r \geq 0$, $\forall\ r \in R$. ∎

To summarize, if $\sum_{r \in R^+} v^r$ empty trailers are available at the CC, then Lemma 3 shows that empty balancing can be achieved with the routes and loaded trailer assignments generated in IGLH_MATCH Steps 1 and 2 without additional tractor miles. Empty trailer balancing can be accomplished, however, with an initial inventory of empty trailers at the CC that may be significantly less than $\sum_{r \in R^+} v^r$ empty trailers and similarly, with a final inventory that is less than $|\sum_{r \in R^-} v^r|$. Specifically, let the total group imbalance, denoted IMB, equal $\sum_{r \in R^+} v^r + \sum_{r \in R^-} v^r$. Then, through appropriate sequencing of certain routes, empty balancing can be achieved without additional tractor miles with an initial inventory of MAX(IMB, 0) empty trailers at the CC and a final inventory of MAX(- IMB, 0). Before proving this, we detail the Step 3 IGLH_MATCH procedure, called *Sequence Routes*, as follows:

Sequence Routes **Step 1:** Let $e^r = X^r - x^r$ be the number of empty trailers assigned to each leg of every route $r \in R$. Partition the set R into the following subsets: $R_I = \{r \mid \sum_{r=1}^{k} v^r = IMB$, r = 1,2,...,k $\in R$, k as small as possible$\}$; $R_0 = \{r \mid v^r = 0, r \in R \backslash R_I\}$; $R_1^+ = \{r \mid v^r = 1, r \in R \backslash R_I\}$; $R_1^- = \{r \mid v^r = -1, r \in R \backslash R_I\}$; $R_2^+ = \{r \mid v^r = 2, r \in R \backslash R_I\}$; and $R_2^- = \{r \mid v^r = -2, r \in R \backslash R_I\}$. Let $|R_1| =$ MIN($|R_1^+|$, $|R_1^-|$) and $|R_2| =$ MIN($|R_2^+|$, $|R_2^-|$) where $|\ S\ |$ denotes the number of elements in set S. Initialize $P = \{R_0\}$.

Sequence Routes **Step 2:** Repeat the following steps $|R_1|$ times: 1) select any route, denoted r_1^+, from R_1^+ and any route, denoted r_1^-, from R_1^-; 2) let r^S denote the *super-route* formed by performing route r_1^- followed by route r_1^+; 3) let $P = P + r^S$; 4) let $R_1^+ = R_1^+ - r_1^+$; and 4) let $R_1^- = R_1^- - r_1^-$.

Sequence Routes **Step 3:** Repeat the following steps $|R_2|$ times: 1) select any route, denoted r_2^+, from R_2^+ and any route, denoted r_2^-, from R_2^-; 2) let r^S denote the *super-route* formed by performing route r_2^- followed by route r_2^+; 3) let $P = P + r^S$; 4) let $R_2^+ = R_2^+ - r_2^+$; and 4) let $R_2^- = R_2^- - r_2^-$.

Sequence Routes **Step 4:** Repeat the following steps until R_1^+ is empty: 1) select any two routes, denoted r_1^+ and q_1^+, from R_1^+ and select any route, denoted r_2^-, from R_2^-; 2) let r^S denote the *super-route* formed by performing route r_2^- followed by routes r_1^+ and q_1^+; 3) let $P = P + r^S$; 4) let $R_1^+ = R_1^+ - r_1^+ - q_1^+$; and 4) let $R_2^- = R_2^- - r_2^-$.

Sequence Routes **Step 5:** Repeat the following steps until R_2^+ is empty: 1) select any two routes, denoted r_1^- and q_1^-, from R_1^- and select any route, denoted r_2^+, from R_2^+; 2) let r^S denote the *super-route* formed by performing routes r_1^- and q_1^- followed by route r_2^+; 3) let $P = P + r^S$; 4) let $R_1^- = R_1^- - r_1^- - q_1^-$; and 4) let $R_2^+ = R_2^+ - r_2^+$.

Sequence Routes **Step 6:** Let r* = 0 and repeat the following steps $|R_I|$ times : 1) select from R_I any route, denoted r_I, with $v^{r_I} < 0$ if one exists, otherwise select any route, denoted r_I, with $v^{r_I} \geq 0$; 2) let r^* denote the super-route formed by performing route r^* followed by route r_I; and 3) let $R_I = R_I - r_I$.

***Sequence Routes* Step 7:** Let $\boldsymbol{P} = \boldsymbol{P} + r^*$.

Lemma 4: Given the routes, the loaded trailer assignments generated in IGLH_MATCH Steps 1 and 2, and the empty trailer assignments $e^r = X^r - x^r$ for each route $r \in \boldsymbol{R}$, minimization of the initial and final empty trailer inventories at the *CC* can be achieved with the route set $\boldsymbol{P}$.

Proof: From the definition of v^r, the absolute value of imbalance for any route $r \in \boldsymbol{R}$, i.e., $|v^r|$, is either 0, 1 or 2. Therefore, sets $\boldsymbol{R}_I$, $\boldsymbol{R}_0$, $\boldsymbol{R}_1^+$, $\boldsymbol{R}_1^-$, $\boldsymbol{R}_2^+$, $\boldsymbol{R}_2^-$ partition set $\boldsymbol{R}$. Furthermore, by design, $\sum_{r \in R_I} v^r = \sum_{r \in R} v^r$ and it follows that $\sum_{r \in R \setminus R_I} v^r = 0$, or equivalently, $\sum_{r \in R_1^+ \cup R_2^+} v^r = \sum_{r \in R_1^- \cup R_2^-} v^r$. Thus, there always exist appropriate routes for selection in *Sequence Routes* steps 2, 3, 4 and 5. Each route $r \in \boldsymbol{P} \setminus r^*$ is balanced and thus, from Lemma 3, requires no inventory of empty trailers at the *CC*. Furthermore, if *IMB* is positive, route r^* requires an initial inventory of *IMB* empty trailers at the *CC* while if *IMB* is negative, route r^* results in a final inventory of $|IMB|$ empty trailers at the *CC*. ■

The *Sequence Routes* procedure is illustrated in Table 5 for two problems: one with a positive group imbalance and another with a negative imbalance. The examples show that through appropriate sequencing and merging of routes, the inventory of empty trailers at the *CC* can be reduced substantially. Specifically, the required inventory of empty trailers at the *CC* can be reduced from 6 to 2 for Problem 1 and from 4 to 0 for Problem 2.

Table 5. *Sequence Routes* Procedure

	Problem 1				Problem 2			
Steps 1 & 2 IGLH_ MATCH routes	Route Imbal- ance†	Number of Empties required at the CC	IGLH_MATCH Step 3 Routes (Sequence Routes)		Route Imbal- ance†	Number of Empties required at the CC	IGLH_MATCH Step 3 Routes (Sequence Routes)	
			Sequence	Imbalance			Sequence	Imbalance
r_1	-1	0	$r_1 \rightarrow r_4$	0	0	0	r_1	0
r_2	0	0	r_2	0	-1	0	$r_2 \rightarrow r_3$	0
r_3	2	2	-	-	1	1	-	-
r_4	1	1	-	-	-2	0	$r_4 \rightarrow r_6$	0
r_5	-1	0	$r_5 \rightarrow r_{10}$	0	-2	0	r_5	-2
r_6	0	0	r_6	0	2	2	-	-
r_7	0	0	r_7	0	0	0	r_7	0
r_8	2	2	r_8	2	-1	0	$r_8 \rightarrow r_9$	0
r_9	-2	0	$r_9 \rightarrow r_3$	0	1	1	-	-
r_{10}	1	1	-	-	0	0	r_{10}	0
Total	2	6		2	-2	4		-2

† : Route imbalance equals the total number of pickups minus the total number of deliveries for that route.

3. COMPUTATIONAL EXPERIENCES

The IGLH_MATCH procedure was tested using both randomly generated data designed to reflect the IGLH operations of large LTL carriers and *real-world* data provided by a large LTL carrier. The randomly generated problems have from 2 to 100 EOL terminals, and for each number of EOL terminals, a set of five different problems are generated. The EOL terminals are categorized as small, medium, or large, with each size equally likely to occur in the market. The required pickups and deliveries at each terminal are determined from the probability mass function shown in Table 6. Based on typical intra-group line-haul operations at various LTL carriers, the maximum number of pickups (deliveries) at any one terminal is assumed to be four. The distances between each pair of terminals and between the terminal and the *CC* were generated so as to preserve the triangle inequality.

Table 6. Probability Mass Function for Pickups and Deliveries

EOL Terminal Size			Small	Medium	Large
Probability of	1	pickup	0.90	0.50	0.30
		delivery	0.75	0.40	0.20
	2	pickups	0.10	0.40	0.40
		deliveries	0.20	0.40	0.30
	3	pickups	0.00	0.10	0.30
		deliveries	0.05	0.20	0.40
	4	pickups	0.00	0.00	0.00
		deliveries	0.00	0.00	0.10

The total number of pickups and deliveries for each problem were not necessarily balanced. This reflects, for example, the fact that the number of deliveries on Monday often exceeds the number of pickups, with the reverse occurring on Friday.

The data representing the IGLH problem of a large LTL carrier contained 27 EOL terminals, with a total of 151 pickups and 156 deliveries.

The purpose of the computational experiments was twofold: 1) to evaluate the quality of the solutions generated by IGLH_MATCH; and 2) to determine both the memory and run time requirements of the IGLH_MATCH procedure.

Implementation

The IGLH_MATCH procedure was implemented in the C programming language and all computational tests were performed on an IBM RS\6000, Model 370 workstation.

The *Match* and *Re-Match* procedures in IGLH_MATCH Step 2 used the nonbipartite weighted matching algorithm of Gabow (1973) with its worst case running time O(n^3), where n is the number of network nodes.

Computational Results

For the smaller test problems, i.e., up to eight EOL terminals, the objective function values of the IGLH_MATCH solutions were compared with the optimal IGLH solution values obtained using a branch-and-bound procedure(IGLH_BB), implemented using IBM's Optimization Subroutine Library (OSL) (Optimization Subroutine Library, 1992). Table 7 shows that the IGLH_MATCH procedure determined optimal solutions for problems containing up to six EOL terminals. For the data sets containing seven and eight EOL terminals, only two of the five test problems could be solved exactly in each case. For these problems, the average gaps between the IGLH_MATCH and exact solutions were 6.44% and 3.59%, respectively. The average median IGLH_MATCH solution time was less than 1/7 seconds, compared to 391.7 seconds required for IGLH_BB.

For larger problems containing nine or more EOL terminals, optimal solutions could not be achieved due to insufficient memory. Hence, the IGLH_MATCH objective function value was compared to a lower bound on the optimal IGLH solution, obtained by solving the linear relaxation of IGLH (IGLH_LP), implemented using OSL. For these problems, Table 7 shows that the IGLH_MATCH procedure typically generates solutions within ten percent of the IGLH_LP lower bound, with the gap decreasing as the number of EOL terminals increases. The IGLH_MATCH solution time never exceeded 10 seconds, while the average median IGLH_LP solution time was 182.9 seconds. For problems containing 100 EOL terminals, over one thousand seconds were required by the IGLH_LP procedure.

Although the LTL carrier's problem was too large to determine an optimal solution, we evaluated the IGLH_MATCH solution by comparing it to the solution generated and used by the carrier. The IGLH_MATCH solution was generated in less than two seconds and its cost was 11% less than that of the carrier.

In conclusion, IGLH_MATCH produces quality solutions for very large IGLH problems in seconds.

Table 7. IGLH_MATCH Computational Results for Randomly Generated Problems

Number of EOL terminals	Median Run Time (sec) †			Solution Gap (%) ‡	
	IGLH_MATCH	IGLH_BB	IGLH_LP	IGLH_MATCH VS. IGLH_BB	IGLH_MATCH VS. IGLH_LP
2	0.0	0.0	0.0	0.00	23.48
3	0.0	0.0	0.0	0.00	19.20
4	0.0	1.0	0.0	0.00	15.38
5	0.0	8.0	0.0	0.00	12.43
6	0.0	155.0	0.0	0.00	7.98
7	0.0	1225.0	0.0	6.44	12.48
8	0.0	1353.0	0.0	3.59	9.44
9	0.0	*	0.5	N/A	11.58
10	0.0	*	1.0	N/A	10.59
15	0.0	*	1.0	N/A	8.93
20	0.0	*	3.0	N/A	12.37
25	1.0	*	5.0	N/A	8.86
30	1.0	*	10.0	N/A	8.24
35	1.0	*	16.0	N/A	6.18
40	1.0	*	26.5	N/A	6.31
45	1.0	*	41.0	N/A	4.81
50	2.0	*	65.0	N/A	6.66
55	3.0	*	93.0	N/A	5.11
60	3.0	*	126.5	N/A	4.26
65	4.0	*	195.0	N/A	5.97
70	4.0	*	255.0	N/A	4.14
75	5.0	*	322.0	N/A	4.82
80	5.0	*	479.5	N/A	4.41
85	6.0	*	568.0	N/A	4.53
90	7.0	*	716.5	N/A	4.45
95	9.0	*	933.5	N/A	4.04
100	9.0	*	1079.5	N/A	4.16
Average	2.30	N/A	182.87	N/A	8.55

† : IBM RS\6000, Model 370 workstation

‡ : Solution Gap (%) = {(IGLH_MATCH solution - IGLH_BB (or IGLH_LP)) / IGLH_BB (or IGLH_LP)} * 100

* : Run time was not available due to insufficient memory.

4. MODEL FLEXIBILITY AND ADAPTABILITY

The core IGLH problem presented and formulated in Section 1 is a simplified version of the IGLH problem faced by LTL carriers. Several practical considerations, such as driver work rules and level of service requirements, are not included in the model formulation (1) - (7). The IGLH_MATCH procedure, however, can easily accommodate such restrictions.

Consider for example, the work rule limiting the total number of hours a driver can be on duty in any given day. This rule can be enforced within the IGLH_MATCH procedure with a simple legality check. That is, since each link in the *Match* and *Re-Match* network corresponds to a route, each route can be evaluated to ensure that its total elapsed time does not exceed the limit. If the limit is exceeded, the link corresponds to an illegal route and its cost is set to infinity. These legality checks can be performed quickly, with a small increase in the total solution run time.

Our computational experiments show that within a few seconds, the IGLH_MATCH procedure can produce solutions to IGLH problems that are larger than those faced by most American LTL carriers. Thus, IGLH_MATCH can provide a tool for real-time scheduling of IGLH operations. Not all required pickups and deliveries are known at the time driver routes and schedules are developed. There is a need, therefore, to alter certain routes and develop new routes while keeping some routes fixed. The IGLH_MATCH procedure fixes a route by removing from consideration the pickups and deliveries it services, i.e., by eliminating the corresponding nodes (and hence, links) in the matching network. An existing route is altered by including the route as a node in the matching network and adding links to every other network node, with link cost set to infinity if the corresponding altered route is illegal. Finally, altered and new routes are determined by applying the IGLH_MATCH procedure to the modified network. Hence, IGLH_MATCH can produce very quickly new IGLH solutions that reflect current pickup and delivery information and previously defined routes.

5. CONCLUSIONS AND FUTURE RESEARCH

IGLH_MATCH, a simple, flexible procedure relying on matching theory, has been shown to produce quality IGLH solutions extremely quickly on a workstation class computer. These IGLH solutions attempt to minimize simultaneously the number of tractor miles traveled and the empty trailer inventory size at the *CC*. Practical considerations concerning work rules and level of service requirements can be modeled easily and incorporated into the IGLH_MATCH solution process with a simple legality check. Furthermore, the IGLH_MATCH procedure is a viable method for real-time solution of IGLH problems: partial solutions can be fixed and re-optimization is extremely fast.

Future work is necessary to quantify further the practical value of the IGLH_MATCH procedure. That is, it is necessary to perform *extensive* computational experiments comparing IGLH_MATCH solutions to those obtained by practitioners solving IGLH problems daily. Through these experiments, the savings associated with the IGLH_MATCH procedure can be quantified.

Bibliography

1. Sheffi, Y. and W. Powell (1985), "Interactive Optimization for LTL Network Design", Center for Transportation Studies Report 85-17, Massachusetts Institute of Technology, Cambridge, Massachusetts.

2. Golden, B. L. and A. A. Assad (1988), "Vehicle Routing : Methods and Studies", **Studies in Management Science and Systems**, Vol. 16, North-Holland, Amsterdam.

3. Magnanti, T.L. (1981), "Combinatorial Optimization and Vehicle Fleet Planning : Perspectives and Prospects", **Networks, Vol. 11**, pp. 179 - 213.

4. Sheffi, Y. and J. Eckstein (1987), "Optimization of Group Line-Haul Operations for Motor Carriers Using Twin Trailers", **Transportation Research Record 1120**, pp. 12-23.

5. Eckstein, J. (1986), "Routing Methods for Twin-Trailer Trucks", Masters Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, Operations Research Center.

6. Gabow, N. H. (1973), "Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs", Ph.D. Thesis, Stanford University, Stanford, California.

7. Optimization Subroutine Library (1992), "Guide and Reference, Release 2", Publication No. SC23-0519-03, IBM Corporation, Kingston, NY.

A stochastic user equilibrium (SUE) path flow estimator for the DEDALE database in Lyon

by Michael G H Bell[1], Caroline M Shield[1], Jean-Jacques Henry[2] and Laurent Breheret[3]

Abstract

The paper sets out a path flow estimator suitable for the DEDALE database in Lyon, where traffic flow measurements are available at 6-minute intervals from 42 Traffic Data Collection Units (permanent counting stations). The estimator assumes that traffic assigns itself to paths according to the logit route choice model and that congestion leading to delay only forms on a link when its capacity is reached. An equivalent convex programming problem is formulated and an iterative solution procedure is set out. The estimation of the dispersion factor in the logit model is discussed, and a column generation method is proposed to avoid the need for path enumeration. A number of propositions are proved.

1. Introduction

Most Advanced Transport Telematics (ATT) systems require up-to-date information about the state of the road network. However, each system collects only a limited range of data. As an example, urban traffic control and EURO-SCOUT dynamic route guidance collect data from different sources (vehicle detectors and floating vehicles) but both require information about the state of the network (travel times and traffic flows). Moreover, ATT systems generally interact with each other, so there may be a requirement for management to resolve conflicts. The need for a *transport supervisor* has therefore been perceived, particularly within the DRIVE program. Such supervisors are being installed in Turin, Southampton and Lyon.

The function of the supervisor is to monitor the state of the transport system as a whole, to facilitate the exchange of data between ATT systems and, where problems arise, to manage transport in such a way as to solve the problem. Typically the ATT systems would be linked by a ring, as illustrated in Fig. 1 which relates to the Salerno pilot project (Di Taranto et al., 1993).

A form of network information that is potentially of use to a wide range of ATT systems is the current set of path flows. Knowledge of this would allow traffic signal coordination to be improved. Knowledge of the link flows and turning movements, which may be derived from the path flows, would allow the control at individual junctions to be improved. Path flows are not, however, directly observable. While link flows may be readily measured, manually or automatically, they provide only indirect information about path flows. In order to relate link to path flows, assumptions must be made about the nature of route choice (referred to as *traffic assignment*). In uncongested networks, it is possible to separate the route choice and path flow estimation problems; in congested networks the two problems are not separable.

There is an extensive literature on the related problem of estimating Origin-Destination (O-D) flows from traffic counts, most of it relating to uncongested networks (see Section 2). O-D flows are an aggregated form of path flows. Conventionally O-D flows are assumed to be fixed in some sense whereas path flows vary according to network conditions. It is possible to estimate path flows given O-D flows by using an appropriate traffic

[1] Transport Operations Research Group, University of Newcastle upon Type, NE1 7RU, UK

[2] Centre D'Etude Et De Recherche Toulouse, 2 Avenue Edouard Belin BP 4025, Toulouse F - 31055, France

[3] SODIT, Technoparc Bât 6 - Voie n°.5 - Labège Innopole - BP 533, 31674 LABEGE Cedex FRANCE

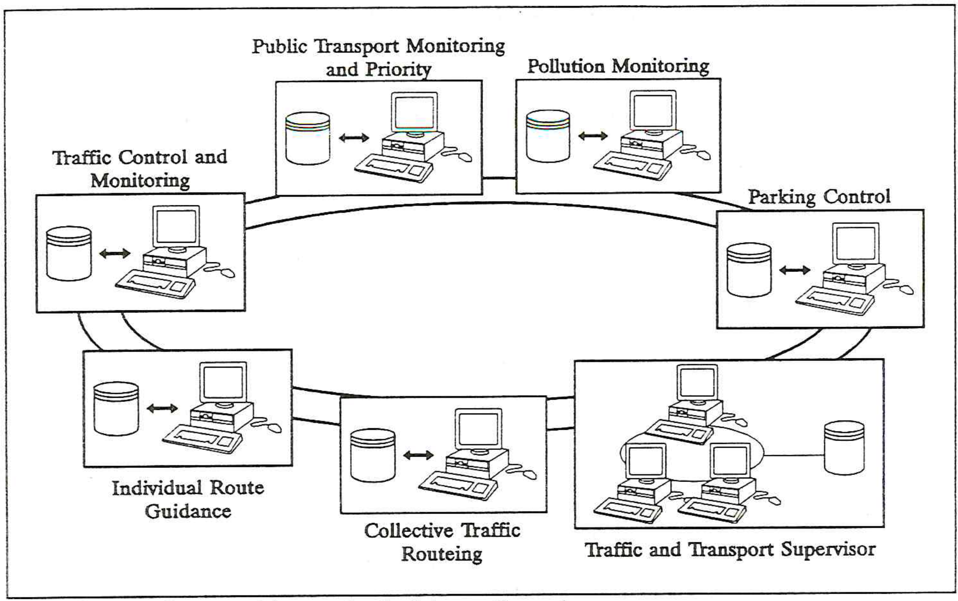

Public Transport Monitoring and Priority
Pollution Monitoring
Traffic Control and Monitoring
Parking Control
Individual Route Guidance
Collective Traffic Routeing
Traffic and Transport Supervisor

Figure 1: System architecture for the Salerno pilot project

assignment model. However, the (same) traffic assignment model is also required to estimate O-D flows from measurements of link flows. If the starting point is measurements of link flows, it seems more direct to estimate path flows directly, which is the approach adopted in this paper.

2. Review of O-D estimation

There are two fundamentally different approaches to O-D estimation. The static approach assumes steady state conditions in the network, and is therefore more suitable for longer time horizons and larger networks. The dynamic approach uses variation in the traffic flows in order to identify input-output coefficients, leading to recursive estimators that are more suitable for shorter time horizons and smaller networks. While O-D estimators tend to be either static or dynamic, it is possible to conceive of hybrid estimators which use time series data to identify local input-output coefficients dynamically and aggregate data to estimate O-D patterns at the network level. The applicability of the static, dynamic and hybrid approaches is illustrated in Table 1.

Table 1: Applicability of static and dynamic estimators

Time horizon **Type of urban network**	**Long term**	**Medium term**	**Short term**
Small with no route choice	Static	Dynamic	Dynamic
Large with no congestion	Static	Hybrid	Hybrid
Large with congestion	Static / Bilevel programming		

When networks offer no route choice or are uncongested, the O-D estimation and the traffic assignment subproblems may be separated, as noted in Section 1. In this case, the relationship between the O-D matrix and the link flows may be represented by a set of linear equations which underlie the following O-D estimators: the maximum entropy and minimum information estimators proposed by Van Zuylen and Willumsen (1980); the generalised least squares estimator proposed by Cascetta (1984) and Bell (1991a); the Bayesian estimator, proposed by Maher (1993), which is identical to the generalised least squares estimator; and a class of maximum likelihood estimators proposed by Spiess (1987).

When networks are congested, the estimation of O-D flows and assignment can no longer be separated. This has led to the proposed use of bilevel programming, whereby the O-D estimation and traffic assignment sub-problems are solved in sequence (see Florian et al., 1991; Yang et al., 1992). A more satisfactory approach makes use of linear programming (see Sherali et al., 1994) to estimate user optimal path flows, which may then be aggregated to yield an O-D matrix.

For junctions or small networks, time series statistical techniques may be applied to identify input-output coefficients. For small networks where there is no route choice, recursive estimation procedures are appropriate (see Cremer and Keller, 1987; Nihan and Davis, 1987; Bell, 1991b). As the size of the network increases, the time lags between inputs and outputs become more variable because of platoon dispersion within the network. This in turn makes the coefficients more difficult to identify. For larger networks which either offer no route choice or are uncongested, a hybrid static-dynamic approach, as formulated by Keller and Ploss (1987), looks attractive. Input-output coefficients for junctions are determined dynamically. These are then used as additional constraints in the static estimation of O-D flows.

The ODIN project of DRIVE I, which studied the applicability of a number of promising static and dynamic O-D estimators to a range of network types (urban and inter-urban, with and without route choice) and produced preliminary recommendations. ODIN also produced a number of methodological advances, notably an event based O-D estimator suitable for very small networks (reported in Bell et al, 1991) and ways to allow for travel time variation in recursive estimation suitable for medium sized networks without route choice (reported in Bell, 1991b).

When short term prediction is required for larger networks, time dependencies must be taken into account. Approaches suggested so far have assumed proportional assignment and are therefore most appropriate for uncongested networks (see Ashok and Ben- Akiva, 1993).

3. A path flow estimator

The estimator proposed in this paper is designed to produce on-line estimates of current link flows and turning movements for the DEDALE network in Lyon from 6-minute flow data emanating from 42 Traffic Data Collection Units (permanent counting stations).
The approach adopted is to estimate path flows under the assumption of Stochastic User Equilibrium (SUE) assignment. Steady state network conditions are assumed. Each link is assumed to have a constant travel time and capacity. When capacity is reached a queue can form leading to delay. The division of trips between alternative paths connecting any O-D pair is on the basis of the logit model, where the path costs consist of travel time and queuing time. The SUE assignment model adopted here is described in Bell (1993b). The logit model parameter governing the degree of dispersion across alternative paths can be estimated on-line. A column generation method is proposed to avoid the specification of all paths.

4. Traffic assignment

User equilibrium (UE) assignment has, since Wardrop (1952), played a dominating role in the literature on traffic assignment for congested networks. Underlying UE assignment are the assumptions that for each O-D pair every link has a cost known to all drivers, all used paths have minimum cost and all paths that have greater than minimum cost are not used.

In urban networks, the most significant component of delay arises at junctions because here capacity is least. This realisation focuses attention on queuing behaviour at junctions. Thus many traffic simulation models concentrate on queuing behaviour and make simplifying assumptions about link travel times. Early versions of the CONTRAM traffic simulator (Leonard et al, 1978) assumed vertical queuing and constant link travel times. The DEDALE estimator takes the same approach and assumes additionally steady state network conditions. This means that traffic on each link is processed at a constant rate, and a queue can only exist on links operating at capacity.

Smith (1987) has proved the Thompson and Payne (1975) result that a UE assignment can be found for networks of the kind assumed here by solving a linear programming problem. The total travel time in the network, excluding queuing time, is minimised subject to link capacity and non-negativity constraints. The dual variables (the Lagrange multipliers associated with the capacity constraints) are equal to the UE delays (see Bell, 1993b, for a proof).

ATT is increasing the number of possibilities for informing drivers of congestion and how to avoid it. These developments are serving to highlight the fact that drivers generally lack perfect information about network conditions, and that therefore one assumption behind UE assignment is significantly violated. This is leading to a resurgence of interest in SUE assignment (see Bell, 1993a).

The form of SUE assignment considered in this paper assumes that vehicles are assigned to routes according to the logit model. Fisk (1980) has shown how such an assignment is the solution to a particular convex programming problem formed from the Smith (1987) linear programming problem by adding a term to the objective function. Bell (1993b) has demonstrated that the dual variables of the Fisk problem are still interpretable as equilibrium delays (this time SUE delays rather than UE delays).

5. Notation

Let:

c_i = the cost of link i
t_i = undelayed travel time for link i
d_i = delay for link i
v_i = the flow on link i
s_i = capacity for link i
q_i = queue for link i
m_i = Lagrange multiplier for link i
h_j = the flow on path j
c_j = the cost of path j
r_k = the count at station k
l_k = Lagrange multiplier for station k

Vector notation is used to represent sets of the above variables. Thus **v** represents $[v_1, v_2, v_3, \dots]$.

6. Representation of link costs

The cost for link i, c_i, has two components; the (constant) undelayed travel time, t_i, and the delay due to queuing, d_i. On the one hand, if $d_i>0$ then $v_i=s_i$. On the other hand, if $v_i<s_i$ then $d_i=0$.

Thus for each link i

$$c_i = t_i + d_i \tag{1}$$

Following Smith (1987), delay is given by the queue divided by the saturation low

$$d_i = q_i / s_i \tag{2}$$

Given that traffic is serviced only during green intervals, there is a difficulty in interpreting d_i, q_i and s_i. In reality of course, the queue will grow during the red interval and decline during the green interval. Hence d_i should be interpreted as the average delay which vehicles expect to encounter and s_i is the average rate at which the queue is serviced. These difficulties of interpretation arise because we are enforcing steady state assumptions on a system which is not steady state.

7. Relationship between path and link flows

Let A be the link-path incidence matrix with elements a_{ij} equal to 1 if link i lies on path j and 0 otherwise. The relationship between the path flows and the link flows may then be expressed as

$$\mathbf{v} = \mathbf{Ah} \tag{3}$$

Further, let **B** be the incidence matrix relating paths to counting stations with elements b_{kj} equal to 1 if path j passes counting station k and 0 otherwise. The relationship between the counts and the path flows may then be expressed as

$$\mathbf{r} = \mathbf{Bh} \tag{4}$$

8. Stochastic user equilibrium

Definition (SUE assignment): A stochastic user equilibrium (SUE) is achieved when the allocation of trips between alternative paths conforms to the following logit model

$$\ln(h_j/h_{j'}) = -\alpha(c_j - c_{j'}) \tag{5}$$

where j and j' are alternative paths connecting the same O-D couple and $\alpha > 0$ is a given parameter. ■

Consider the following problem

P_1: Minimise $\Sigma_j\, h_j\, (\ln h_j - 1) + \alpha\, \mathbf{t}^T\mathbf{Ah}$ with respect to **h** subject to $\mathbf{r} = \mathbf{Bh}$, $\mathbf{s} \geq \mathbf{Ah}$, $\mathbf{h} \geq \mathbf{0}$

The first term in the objective function has the effect of spreading the assignment. The following Lagrangian equation can be formed

$$£ = \Sigma_j\, h_j(\ln h_j - 1) + \alpha\, \mathbf{t}^T\mathbf{v} + \mathbf{l}^T(\mathbf{r} - \mathbf{Bh}) + \mathbf{m}^T(\mathbf{s} - \mathbf{v}) \tag{6}$$

where $\mathbf{v} = \mathbf{Ah}$. The Kuhn-Tucker conditions

$$\ln h_j + (\alpha\, \mathbf{t} - \mathbf{m})^T\mathbf{a}_j - \mathbf{l}^T\mathbf{b}_j = 0 \tag{7}$$

and

$$r_k = \Sigma_j\, b_{kj}\, h_j \tag{8}$$
$$s_i - v_i \geq 0 \tag{9}$$
$$m_i \leq 0 \tag{10}$$
$$m_i(s_i - v_i) = 0 \tag{11}$$

are satisfied at the optimum. The form of (7) ensures that $h_j > 0$. Equation (7) reduces to

$$\ln h_j = -\alpha\, t_j + m_j + l_j \tag{12}$$

where l_j is the sum of counting station Lagrange multipliers l_k along path j, t_j is the undelayed path travel time and m_j is the sum of link capacity Lagrange multipliers m_i also along path j.

Proposition 1 proves that, for any link, $m_i = -\alpha d_i$ is necessary and sufficient for SUE.

Proposition 1 (SUE delays for link constraints): P_1 yields a SUE assignment if and only if the Lagrange multipliers $\mathbf{m}$ associated with the flow constraints $\mathbf{s} \geq \mathbf{v} = \mathbf{Ah}$ are equal to $-\alpha\mathbf{d}$, where $\mathbf{d}$ is the vector of SUE link delays.

Proof:

(i) Suppose $v_i = s_i$. There are then two possibilities. The first is that the demand for link i is exactly s_i. Nothing would be gained, in terms of a reduction in the objective function value, by shifting trips from paths not using i to paths using i. Therefore, a small increase in s_i leads to $v_i < s_i$, so refer to case (ii). The second possibility is that the demand for link i exceeds s_i, so the constraint binds. Link i is therefore the bottleneck on path j. A small increase in s_i would allow the objective function to be reduced by shifting some trips from paths not using link i to paths using link i. To minimise the objective function, the shift will be made to minimise the objective function value. Suppose that a small shift of δ trips is made from path j' to path j. Ignoring terms in δ^2 and higher powers of δ, $\ln(h+\delta) = \ln h + \delta/h$, so the change in objective function value is given by

$$d£ = \delta \ln(h_j/h_{j'}) + \delta\alpha(t_j - t_{j'}) \qquad (13)$$

Hence $m_i = \ln(h_j/h_{j'}) + \alpha(t_j - t_{j'})$. But according to SUE, trips are allocated to paths according to the logit model as in (5). As link i is the bottleneck on path j and path j' has no active bottleneck after the shift, we have

$$\ln(h_j/h_{j'}) = -\alpha(t_j + d_i - t_{j'}) \qquad (14)$$

from which we deduce that $m_i = -\alpha d_i$ is necessary for SUE.

(ii) Suppose $v_i < s_i$. By the complimentary slackness conditions (9) to (11) $m_i = 0$. Also $d_i = 0$ as link i is operating below capacity.

Thus $m_i = -\alpha d_i$ is necessary and sufficient for SUE. ■

There is an issue relating to the uniqueness of the equilibrium delays. The objective function of P_1 can be shown to be strictly convex in path flows, so the path flows are uniquely defined. The link flows are therefore uniquely defined by (3). This implies that the set of links for which $m_i = d_i = 0$ is also uniquely defined. Proposition 2 proves that the linear independency of the capacity constraints is necessary and sufficient for unique equilibrium delays.

Proposition 2 (uniqueness of SUE delays): The SUE delays are unique if and only if all the constraints that bind are linearly independent.

Proof: Drop the links (rows) for which $m_i = d_i = 0$ from $\mathbf{A}$ to produce $\mathbf{A}'$. Writing (7) in matrix notation

$$\mathbf{h}^T = \exp(-\alpha\mathbf{t}^T\mathbf{A} + \mathbf{m}^T\mathbf{A}' + \mathbf{l}^T\mathbf{B}) = \exp(-\alpha\mathbf{t}^T\mathbf{A} + \phi^T\mathbf{D}) \qquad (15)$$

where $\phi^T = [\mathbf{m}^T,\mathbf{l}^T]$ and $\mathbf{D}^T = [\mathbf{A}'^T,\mathbf{B}^T]$. Thus

$$d\mathbf{h}/d\phi = \mathbf{HD}^T \qquad (16)$$

where $\mathbf{H}$ is a diagonal matrix whose jth diagonal element is equal to h_j. Let $\Theta^T = [\mathbf{s}'^T,\mathbf{r}^T]$, where $\mathbf{s}'$ is a vector of capacity constraints that bind at the SUE solution. By the Chain Rule of differentiation

$$d\Theta/d\phi = (d\Theta/d\mathbf{h})(d\mathbf{h}/d\phi) = \mathbf{DHD}^T \qquad (17)$$

Suppose ϕ is a solution to

$$\Theta = \mathbf{D}\exp(-\alpha\mathbf{A}^T\mathbf{t} + \mathbf{D}^T\phi) \qquad (18)$$

Consider ϕ' close to ϕ. Then by a first order approximation

$$\mathbf{DHD}^{\mathrm{T}}(\phi - \phi') = \Theta - \mathrm{D}\exp(-\alpha \mathbf{A}^{\mathrm{T}}\mathbf{t} + \mathbf{D}^{\mathrm{T}}\phi') \tag{19}$$

Suppose now that ϕ' is also a solution to (18) so that the right hand side of (19) is zero. Equation (19) implies that $\phi = \varphi'$ if and only if $\mathbf{DHD}^{\mathrm{T}}$ is non-singular. This is only the case if the rows of $\mathbf{D}$ are linearly independent, which in turn is only the case if all the constraints are linearly independent. Conversely, if all the constraints are linearly independent, $\mathbf{DHD}^{\mathrm{T}}$ is non-singular and $\phi = \phi'$ if $\Theta = \mathrm{D}\exp(-\alpha\mathbf{A}^{\mathrm{T}}\mathbf{t} + \mathbf{D}^{\mathrm{T}}\phi')$. ■

9. Solution by iterative balancing

A SUE assignment for a steady state store-and-forward network of the kind described in this paper can be readily found by solving an iterative balancing problem. By rearrangement of (12) we obtain

$$\begin{aligned} h_j &= \exp(-\alpha t_j + m_j + l_j) \\ &= \exp(-\alpha t_j)\ \pi_{i \text{ on } j}\ M_i\ \pi_{k \text{ on } j}\ L_k \end{aligned} \tag{20}$$

where $L_k = \exp(l_k)$ is a factor for counting station k, $M_i = \exp(m_i)$ is a factor for link i, t_j is the undelayed travel time for path j, and m_j and l_j are the sums of m_i and l_k respectively along path j. Factor L_k is calculated so that (4) holds, while factor M_i is calculated so that (9) to (11) hold. The following algorithm finds the values of the Lagrange multipliers satisfying the Kuhn-Tucker conditions.

```
Algorithm A_0 (to find the SUE assignment)
Step 1 (initialisation)
    $M_i = 1$ for all links i
    $L_k = 1$ for all counting stations k
Step 2 (iteration)
    Repeat the following until convergence
        For each i calculate
            $\beta = s_i / \Sigma_j a_{ij} \exp(-\alpha t_j)\ \pi_{i \text{ on } j} M_i\ \pi_{k \text{ on } j} L_k$
            $M_i = \min[1, \beta M_i]$
        For each k calculate
            $\beta = r_k / \Sigma_j b_{kj} \exp(-\alpha t_j)\ \pi_{i \text{ on } j} M_i\ \pi_{k \text{ on } j} L_k$
            $L_k = \beta L_k$

Step 3 (output link flows and delays)
    For each j calculate
        $h_j = \exp(-\alpha t_j)\ \pi_{i \text{ on } j} M_i\ \pi_{k \text{ on } j} L_k$
    For each i calculate
        $v_i = \Sigma_j a_{ij} h_j$
        $d_i = -(\ln M_i)/\alpha$
```

Note that the algorithm ensures that $M_i \leq 1$ so that $m_i \leq 0$.

Proposition 3 (convergence of iterative balancing): The above algorithm converges to the solution of P_1 provided a feasible solution exists.

Proof: Consider the Lagrangian equation given in (6). The Saddlepoint Theorem says that at the optimum this is minimised with respect to the primal variables (link flows $\mathbf{v}$) and maximised with respect to the dual variables (the Lagrange multipliers $\mathbf{l}$ and $\mathbf{m}$). The algorithm maximises £ with respect to the dual variables. If $v_i > s_i$ then £ is increased by reducing m_i. However, this also reduces v_i, so m_i should be reduced until $v_i = s_i$. If $v_i < s_i$ then £ is increased by increasing m_i until either $v_i = s_i$ or $m_i = 0$. Likewise, if $\mathbf{b}_k^{\mathrm{T}}\mathbf{h} > r_k$ (where $\mathbf{b}_k$ is the kth row of $\mathbf{B}$ and $\mathbf{b}_k^{\mathrm{T}}\mathbf{h}$ is therefore the sum of the path flows passing counting station k) then £ is increased by reducing l_k until $\mathbf{b}_k^{\mathrm{T}}\mathbf{h} = r_k$. If $\mathbf{b}_k^{\mathrm{T}}\mathbf{h} < r_k$ then £ is increased by increasing l_k until $\mathbf{b}_k^{\mathrm{T}}\mathbf{h} = r_k$. As these changes to m_i and l_k exactly describe Step 2 of the algorithm, each iteration results in an increase in £. Iterations continue until the Kuhn-Tucker conditions are satisfied. When the Kuhn-Tucker conditions are satisfied, no further increases in £ are possible and P_1 is solved. ■

The requirements for uniqueness of the balancing factors M_i and L_k have been set out in Proposition 2.

10. Determination of α and column generation

Parameter α determines the sensitivity of assignment to path cost. As α increases, the importance of the second part of the objective function increases. In the limit, the assignment tends to UE. As α tends to zero, the driver preference for lesser cost paths disappears. As α increases $\mathbf{t}^T\mathbf{v}$ reduces monotonically (proof of this is along the lines of Erlander et al., 1979). Thus a suitable way to estimate α would be to sample the flows on a sub-set of links. From the sampled link flows and knowledge of the undelayed link travel times, a sampled partial total undelayed travel time can be calculated. Parameter α can then be determined so that sampled and fitted partial totals agree.

Algorithm A_0 assumed that the paths have been enumerated before hand. In complex networks, this is a chore to be avoided. It is possible, however, to generate paths in an iterative way using an appropriate shortest path algorithm, such as Dijkstra's algorithm, as is done in the Frank-Wolfe approach to solving the UE problem. This is referred to as *column generation* because the columns of the link-path incidence matrix **A** are generated as the iterations progress (see Bazaraa et al, 1990). This leads to the following algorithm.

```
Algorithm A_1 (to find the SUE assignment by column generation)
Step 1 (initialisation)
        Set $\mathbf{d} = \mathbf{0}$ (set SUE delays to zero)
        Set A to have no columns (no paths currently exist)
Step 2 (build shortest path trees)
        Set $\mathbf{c} = \mathbf{t} + \mathbf{d}$
        Build fastest path trees using $\mathbf{c}$
        If no new paths generated, stop
        Add new paths to A

Step 3 (determine SUE delays)
        Run $A_0$
        Output $\mathbf{d}$, $\mathbf{h}$ and $\mathbf{v}$
        Return to Step 2
```

In the first few iterations, there may be insufficient capacity in the paths generated to cater for the demand. This would result in infinite delays at the bottlenecks. Although A_0 cannot converge under these conditions, it will indicate which links are the bottlenecks after a few iterations through the presence of vanishing multipliers M_i. For those links, delay can be set to high values before returning to Step 2, thereby causing new paths to be generated.

11. Conclusions

The paper sets out a path flow estimator suitable for the DEDALE database in Lyon, where traffic flow measurements are available at 6-minute intervals from 42 Traffic Data Collection Units (permanent counting stations). The estimator assumes that traffic assigns itself to paths according to the logit route choice model and that congestion leading to delay only forms on a link when its capacity is reached. An equivalent convex programming problem is formulated and an iterative solution procedure is set out. The estimation of the dispersion factor in the logit model is discussed, and a column generation method is proposed to avoid the need for path enumeration.

A number of propositions are proved. Proposition 1 asserts the equivalence between the capacity constraint Lagrange multipliers and SUE delays. Proposition 2 asserts that, while path flows and link flows will be unique, SUE delays will only be unique if the capacity constraints are linearly independent. Proposition 3 asserts that the iterative balancing method may be used to solve for the Lagrange multipliers and therefore for path flows, link flows and SUE delays.

The next step is to test the estimator against a more approach using the conventional entropy-based O-D estimator (OEDIPE) and an assignment model (DAVIS).

Acknowledgements

The authors are grateful to the DRIVE programme of the European Commission for funding the research reported in this paper.

12. References

Ashok, K. & Ben-Akiva, M.E. (1993) Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems. Proceedings of the 12th International Symposium on the Theory of Traffic Flow and Transportation, Elsevier: Amsterdam, 465-484.

Bazaraa, M.S., Jarvis, J.J. & Sherali, H.D. (1990) Linear Programming and Network Flows. John Wiley: New York.

Bell, M.G.H. (1991a) The estimation of generalised least squares by constrained generalised least squares. Transportation Research, Vol. 25B, 13-22.

Bell, M.G.H. (1991b) The real-time estimation of origin-destination flows in the presence of platoon dispersion. Transportation Research, Vol. 25B, 115-125.

Bell, M.G.H., Inaudi, D., Lange, J. & Maher, M. (1991) Techniques for the dynamic estimation of O-D matrices in traffic networks. Proceedings of the DRIVE Conference, Brussels, 1040-1056.

Bell, M.G.H., Lam, H.W.K., Ploss, G. & Inaudi, D. (1993a) Stochastic user equilibrium assignment and iterative balancing. Proceedings of the 12th International Symposium on Transportation and Traffic Theory, Berkeley, July.

Bell, M.G.H. (1993b) Stochastic user equilibrium assignment in networks with queues. Submitted to Transportation Research B.

Cascetta, E. (1984) Estimation of trip matrices from traffic counts and survey data. Transportation Research, Vol. 18B, 289-299.

Cremer, M. & Keller, H. (1987) A new class of dynamic methods for the identification of origin-destination flows. Transportation Research, 21B, 117-132.

Erlander, S., Nguyen, S. & Stewart, N.F. (1979) On the calibration of the combined distribution and assignment model. Transportation Research, Vol. 13B, 259-267.

Fisk, C. (1980) Some developments in equilibrium traffic assignment. Transportation Research, Vol. 14B, 243-255.

Florian, M. & Chen, Y. (1991) A bilevel programming approach to estimating O-D matrix by traffic counts. Report CRT-750, Centre de recherche sur les transports, Montreal.

Leonard, D.R., Tough, J.B. & Baguley, P.C. (1978) CONTRAM - A traffic assignment model for predicting flows and queues during peak periods. TRRL Laboratory Report LR841.

Maher, M.J. (1983) Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. Transportation Research, Vol. 17B, 435-447.

Nihan, N.L. & Davis, G.A. (1987) Recursive estimation of origin-destination matrices from input/output counts.

Transportation Research, Vol. 21B, 149-163.

Sherali, H.D., Sivanandan, R. & Hobeika, A.G. (1994) A linear programming approach for synthesising origin-destination trip tables from link traffic volumes. Transportation Research, In Press.

Smith, M.J. (1987) Traffic control and traffic assignment in a signal-controlled network with queuing. Proceedings of the 10th Symposium on Transportation and Traffic Theory, MIT, July, 61-77.

Spiess, H. (1987) A maximum likelihood model for estimating origin-destination matrices. Transportation Research, Vol.21B, 395-412.

Thompson, W.A. & Payne, H.J. (1975) Traffic assignment on transportation networks with capacity constraints and queuing. Paper presented at the 47th National ORSA/TIMS North American Meeting.

Van Zuylen, H.J. & Willumsen, L.G. (1980) The most likely trip matrix estimated from traffic counts. Transportation Research, Vol. 14B, 281-293.

Wardrop, J.G. (1952) Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers II(1), 325-378.

Yang, H., Sasaki, T., Iida, Y. & Asakura, Y. (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. Transportation Research, Vol. 26B, 417-434.

Implementation and Solution of a Large Asymmetric Network Equilibrium Model *

Stanislaw Berka
David E. Boyce

URBAN TRANSPORTATION CENTER
University of Illinois at Chicago
1033 West Van Buren Street Suite 700 South
Chicago, Illinois 60607
Voice (312) 996-4820 Facsimile (312) 413-0006

May 25, 1994

Abstract

An asymmetric, static user-optimal route choice model is solved in order to generate link travel times for a road network. The delay functions utilized are based on state-of-the-art traffic engineering research. The network creation process and the implementation of the model is described. Results are presented and discussed.

1 Introduction

The ultimate objective of this study is to generate link travel times for a large-scale real-world transportation network for ADVANCE, a dynamic route guidance system. Link travel time estimates are required by time-of-day periods. The method selected for this task is the static asymmetric route choice model. The model is asymmetric because of the interactions among flows and delays within each intersection. In the literature known to the authors, the only documented models of this type were applied to small and medium-size networks, which also did not reflect the complexity of real networks; see Meneguzzer et al (1990), Boyce et all (1989b), Said et al (1993) and Nagurney (1984).

This study utilizes a realistic network with all the variety of intersection configurations and types of transportation facilities including freeways occurring in the test area, the area in which the route guidance system is being deployed. The reasons that there is no study of this kind documented in the literature may be several. First, the cost of obtaining the data and computing the results for this kind of model may be too high compared with the benefits. For the implementation of a route guidance system however, detailed travel time estimates are needed by turning movement, which justifies the higher cost. The link travel times are the initial bases for planning reasonable routes. Although such systems are designed to obtain travel time data from participating vehicles (Boyce et al, 1990), obtaining travel time data in this way may require the system to operate a very long time, even for several thousand participating vehicles (see Hicks et al, 1992). Therefore, there is need for initial travel time estimates.

Another reason may be the current discrepancy between the realistic link delay functions applied in the model and the solution properties of the route choice problem. Theoretical studies indicate that the conditions for existence and uniqueness of the equilibrium solution may be violated for this type of functions. However, computational studies indicate that the solution obtained is unique (Meneguzzer et al, 1990). The third possible reason is the lack of input data required for this kind of model, especially detailed intersection geometry data. The cost of collecting and coding this data for this study was relatively high.

The advantage of applying the model to a route guidance system is that once the system operates, it should be relatively easy in the future to obtain the large amount of measured travel time data to validate the model.

*Prepared for presentation at the TRISTAN 2 Conference, Capri - ITALY, June 1994.

The specific objectives of this study can be outlined as follows.

1. Adapting delay models from the literature, or developing new models when none is available for use as link delay functions, for each type of the road facility and each type of the intersection occurring in the test area.
2. Verifying models and their applicability for route choice problems.
3. Developing a methodology for collecting large amounts of detailed transportation network data, designing the database, coding and testing the data.
4. Extending existing codes for an asymmetric route choice model to solve a problem for a large network. The test area, located in the north and northwest Chicago region, covers approximately 300 square miles and consists of nearly 8000 links and more than 2500 nodes.
5. Solving the model by time of day and generating the travel time estimates to be used within the ADVANCE Project.

Throughout the paper, turning movements are described as LT, TH and RT, which stand for left-turn, through and right-turn movements. For convenience, the references to the 1985 Highway Capacity Manual (Transportation Research Board, 1985) are denoted as 1985 HCM.

The paper begins by presenting the background of the asymmetric user-optimal route choice problem with the conditions of existence and uniqueness of the solution. The next section presents the turning-movement specific travel time functions used in the study. Then, the study network, trip tables and the process of creating them is briefly described, followed by a discussion of the implementation of the model. The last section includes some conclusions and observations on the results obtained in the study.

2 Problem Statement and Background

Route choice models have a long tradition and large literature. A good overview and classification of the models can be found in Said et al (1993) or Patriksson (1991). Route choice problems can be classified according to different criteria: route choice principle; time period; and source of travel demand. The model described in this paper is a static, deterministic user-optimal (UO) route choice problem with fixed demand. The optimization formulation of this problem is presented below, followed by the variational inequalities formulation for the asymmetric case.

Consider a network consisting of L links. The vector

$$\mathbf{f} = (f_1, \ldots, f_L) \tag{1}$$

represents the ***link flow pattern*** in the network, where f_a is the flow of vehicles on link a expressed in vehicles/hour. The cost function

$$\mathbf{c} = (c_1(\mathbf{f}), \ldots, c_L(\mathbf{f})) \tag{2}$$

associates with any given flow pattern a *link cost pattern*, where c_a is the travel cost function of link a. The user-optimal route choice problem consists of assigning a given *trip pattern* to the network, so that Wardrop's principle is satisfied: drivers choose their routes so that the travel time on all routes that are actually used between each origin and destination are equal and not more than the travel time on any unused route (Wardrop, 1952). The trip pattern is defined by the number of trips per hour from each origin zone to each destination zone T_{ij}; in this study, $\mathbf{T} = (T_{11}, \ldots, T_{ij}, \ldots, T_{IJ})$ is given as a fixed matrix.

If each cost function depends on the flow on the same link only, the static fixed demand UO problem can be stated as follows (Beckmann et al, 1956):

$$\text{minimize} \quad \sum_a \int_0^{f_a} c_a(x)dx \tag{3}$$

$$\text{subject to} \quad \sum_{r \in R_{ij}} h_r = T_{ij} \quad \forall i, j \tag{4}$$

$$h_r \geq 0 \quad \forall r \in R \tag{5}$$

where

$$f_a = \sum_{r \in R} \delta_{ar} h_r \quad a = 1. \ldots . L, \tag{6}$$

and h_r = flow on route r
R_{ij} = set of routes from origin i to destination j
T_{ij} = number of trips per hour from origin zone i to destination zone j
δ_{ar} = 1 if the link a belongs to the route r, and 0 otherwise

Problem (3)-(6) is called *diagonal* because the Jacobian matrix (7) of the delay function **c**,

$$J = \left[\frac{\partial c_i}{\partial f_j} \right], \quad i = 1, \ldots, L; \quad j = 1, \ldots, L \tag{7}$$

is diagonal. The Karush-Kuhn-Tucker necessary optimality conditions for this problem are equivalent to Wardrop's principle commonly used within the user-optimal route choice literature as the assumption about drivers' route choice behavior.

In this study a version of problem (3)-(6) with an asymmetric Jacobian matrix (7) is considered. The asymmetric case has been studied rather extensively. Abdulaal and LeBlanc (1979) proved that the mathematical programming formulation for the problem does not exist if matrix is asymmetric. However, Smith (1979) formulated the problem as a system of inequalities, which is a *variational inequality problem*, as pointed out by Dafermos (1980). The variational inequality formulation of the problem and these conditions are presented next

The variational inequality formulation of a UO route choice problem can be stated as:

$$\text{find} \quad \mathbf{f}^* \quad \text{such that} \quad \mathbf{c}(\mathbf{f}^*) \cdot (\mathbf{f} - \mathbf{f}^*) \geq 0, \quad \forall \mathbf{f} \in F \tag{8}$$

where F is a set of feasible link flows satisfying the constraints (4)-(6). For alternative variational inequality formulations of the UO problem see Fisk and Boyce (1983). The conditions for the existence and uniqueness of the solution of (8) were partially obtained by Smith (1979), but later completed and extended by Dafermos (1980) and Dafermos and Nagurney (1984a, 1984b), who applied the theory of VIP originally developed for infinite-dimensional problems in physics.

A sufficient condition for problem (8) to have a solution is that the cost function **c** is continuous and F is a closed convex set. A sufficient condition for the solution to be unique is that the cost function **c** is *strictly monotone*, that is:

$$[\mathbf{c}(\mathbf{f}) - \mathbf{c}(\mathbf{f}')] \cdot (\mathbf{f} - \mathbf{f}') > 0 \quad \forall \mathbf{f}, \mathbf{f}' \in F. \tag{9}$$

The condition of strict monotonicity is satisfied if and only if Jacobian matrix (7) of the cost function **c** is positive definite. Matrix J is positive definite if it satisfies the following conditions (see, for example, Stewart, 1973):

$$\frac{\partial c_i(\mathbf{f})}{\partial f_i} > 0, \quad \forall i \tag{10}$$

$$\frac{\partial c_i(\mathbf{f})}{\partial f_i} > \frac{1}{2} \sum_{j \neq i} \left| \frac{\partial c_i(\mathbf{f})}{\partial f_j} + \frac{\partial c_j(\mathbf{f})}{\partial f_i} \right|, \quad \forall i \tag{11}$$

Condition (10) means that the cost on each link is strictly increasing with the flow on that link. The second condition is satisfied when the link cost depends mainly on the flow on the same link. Note that in the standard route choice problem when

$$\mathbf{c}(\mathbf{f}) = [c_1(f_1), \ldots, c_L(f_L)] \tag{12}$$

condition (11) is satisfied by definition.

In the case of a transportation network, the Jacobian matrix usually has a block-diagonal structure:

$$J = \begin{bmatrix} J_1 \ldots\ldots 0 \ldots\ldots 0 \\ 0 \ldots\ldots J_k \ldots\ldots 0 \\ 0 \ldots\ldots 0 \ldots\ldots J_I \end{bmatrix} \tag{13}$$

where J_k is the Jacobian matrix of the derivatives of travel costs for intersection k and I is the total number of intersections in the network. The reason that the Jacobian has this structure is that the delay at an intersection usually depends only on the flows approaching the same intersection. Smith (1982) proved that in this case the Jacobian is positive definite if and only if each of the blocks on the diagonal is positive definite. The travel time functions used in this study generate a Jacobian that is block-diagonal; however, according to Meneguzzer (1990), conditions (11) are not guaranteed to be satisfied by the class of cost functions applied in this study. The question arises whether the sufficient conditions for the uniqueness of the solution are too strict. Computational results obtained by Meneguzzer (1990) suggest that a unique solution does exist. The question of weaker sufficient conditions remains open.

To solve problem (8) the so-called *diagonalization algorithm* or nonlinear Jacobi is used (see, for example, Meneguzzer, 1990 or Harker and Pang, 1990). The algorithm consists of solving a sequence of nonlinear approximations of the problem, each of which is a nonlinear optimization problem which is solved using the Frank-Wolfe (1956) algorithm. Subproblems are created by *fixing* non-diagonal flows at the level from last iteration, so that within each subproblem each link delay is a function of the flow on the same link only. Dafermos (1982) proved that the diagonalization algorithm converges if the Jacobian matrix is positive definite and loosely speaking "not too asymmetric". These two conditions are only sufficient conditions, so the algorithm may still converge even though they are not satisfied, as appears to be the case in our study. Section 6 presents the implementation of the algorithm.

3 Link Travel Time Functions

This section presents mathematical functions used within a route choice model to estimate link travel times for given flow rates. The choice of the delay functions involves several criteria: (a) the desired mathematical properties of the function to satisfy the conditions for a unique solution of the route choice model; (b) the cost and limited availability of road data; (c) computational effort required by the model and (d) the desired accuracy of the travel time estimates generated by the model. One of the goals of this study was to model travel times by turning movement. Analytical functions are preferred to regression-based models, because the former generate reasonable estimates over a much wider range of input flows and other parameters. Criteria (a) and (b) above exclude many highly detailed traffic engineering models. Criterion (c) excludes simulation models, which are suitable only for small networks.

Delay functions selected for this study can be classified by road type and intersection type. First, delay functions for arterials and collectors are presented (Section 3.1) and then for freeway and tollway related facilities (Section 3.1.4). Table 3 presents a breakdown of intersections by the type of control; Table 2 presents the classification of links by the facility type.

Travel times are modeled for turning movements whenever the difference among them is significant. The delay functions are generally based on the Highway Capacity Manual (Transportation Research Board, 1985, 1993). Some of these models are extended to obtain better estimates (e.g. the model for left-turn capacity at signalized intersections); new models were developed for freeways and freeway-related facilities since those proposed in HCM cannot be used for a wide range of traffic conditions.

Time-dependent delay functions (Kimber and Hollis, 1979) are used to incorporate queuing effects at intersections in oversaturation conditions. These functions are defined for any non-negative volume-to-capacity ratio; for high volumes they are asymptotic to the deterministic model that best describes the delay in heavily oversaturated conditions. See Figure 4 for an example of a time-dependent delay function. These functions are also continuous and strictly increasing, which is required by the sufficient conditions for the existence and uniqueness of the solution of the model stated in Section 2. Some of these functions were adapted from the literature (for example, for signalized intersection or for freeways), while others were derived using the transformation technique developed by Kimber and Hollis (for example for priority intersections and all-way-stop intersections).

The first term of any link travel time function is *cruise time*, which is the time needed to traverse the road segment when no delay occurs at downstream intersections. The second term is the delay experienced by the vehicle at the downstream intersection and is called *intersection delay*. Models for each of these two terms require usually two stages: (a) the analysis of the capacity under prevailing traffic conditions, and (b) the calculation of the delay.

In all models presented in this paper, capacities, saturation flows and volumes are expressed in actual vehicles per hour (vph) unless stated otherwise. The implementation of the delay functions presented in following sections is described in Section 6.

3.1 Arterials and Collectors

For purpose of intersection delay analysis, intersections are classified into several categories according to the type of *intersection control*, the *intersection layout* and the *approach geometry*. Approaches to signalized intersections are assigned a category according to lane designation and intersection geometry. However, actual lane use may be different than the lane designation. An example is an approach with the left lane designated as a LT/TH lane. In the case when the LT movement flow exceeds or is close to LT capacity, the TH drivers try to use the right lane to minimize their delay, and eventually the left lane functions as an exclusive LT lane. The actual lane usage is determined based on the lane flow analysis which estimates flows by turning movement on each approach lane. The analysis is based on the assumption that drivers choose their lane to minimize their delay.

3.1.1 Cruise Time

Cruise time is determined based on the length of the link and the free flow travel time which, according to 1985 HCM, is also a function of the link length. The capacity of an arterial/collector link is practically always higher than the downstream intersection capacity and is not involved in the cruise time estimation.

3.1.2 Signalized Intersections

Signalized intersection models can be used for *intersection design*, in which the problem may consist of selecting the type of control, geometry of the facility or phasing plan (for signalized intersections). The literature on this topic is rather large. Some models optimize all of the above factors (for example Improta and Candarelle, 1984; Rouphail and Radwan, 1989), while others concentrate on cycle length and green split optimization (see Webster and Cobbe, 1966, Webster, 1958, Boyce et al, 1989a). See also Said et al (1993). However, in this study the intersection analysis is applied to generate intersection delay estimates.

Delay models for signalized intersections consist of three modules: (a) saturation flow analysis; (b) signal timing procedure; (c) the delay function itself. Because of the mutual dependencies between the inputs and outputs of the modules (e.g. saturation flows are inputs for the signal setting module, while its output is an input to the saturation flow module), an iterative procedure is required to obtain consistent results. This procedure is described in Section 6. The following sections describe each of the above modules of signalized intersection analysis.

Saturation Flow Saturation flow is a capacity of the approach, lane or lane group under prevailing traffic conditions if the approach received all of the time green signal. It is used together with the length of the actual green time to calculate the capacity of a link. The procedure chosen for this study is a synthesis of the 1985 HCM and Australian method (see Akcelik, 1988a). Conceptually, saturation flow can be defined as:

$$S_a = w(\mathbf{S}', \mathbf{f}', \mathbf{g}, \mathbf{p}) \quad a = 1, \ldots, L \tag{14}$$

where S_a = saturation flow of link a

$\mathbf{S}'$ = vector of saturation flows discharging into the same intersection

$\mathbf{f}'$ = vector of vehicle flows discharging into the same intersection

$\mathbf{g}$ = vector of signal setting parameters at the intersection

$\mathbf{p}$ = vector of other parameters including the intersection geometry

The relationship between the saturation flow of a permissive LT movement and the opposing flow is the most complex of all turning movements. Figure 1 shows an example of such a relationship. In the figure, f^{LT} is the LT flow, g/C is the green time-to-cycle length ratio and f is the total flow on subject approach.

Computationally, the function w is an iterative procedure rather than a function in a closed form. To start the procedure, some default values of saturation flows and signal setting parameters are assumed.

Signal Timing The main parameters describing how the traffic signals operate are: *phase pattern*, *cycle length* and *green split*. The phase pattern defines which movements are allowed in each part of the cycle. In this study it is fixed for each intersection class defined in terms of approach geometry, the number and the designation of approach lanes. The cycle length and the green split are modeled using the Webster (1958) procedure described briefly below. The procedure is used to determine the signal settings similar to those used for real intersections, rather than optimizing these settings. Moreover, the majority of the signals in the study area are actuated signals, which adjust the settings according to existing traffic patterns. Thus, the procedure described below should imitate those actuated signal controllers, which

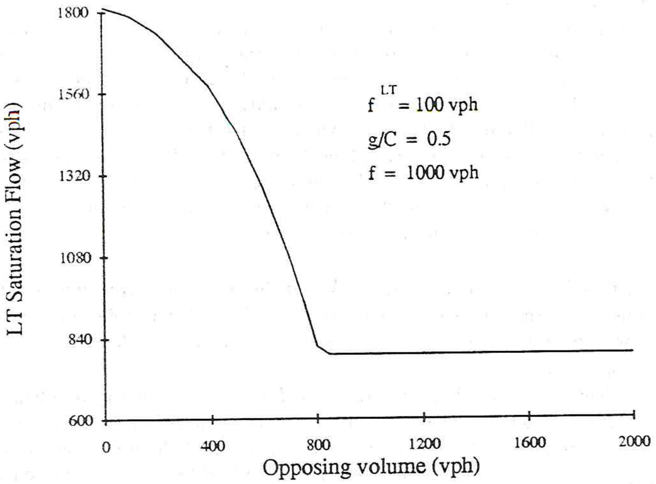


Figure 1: Saturation flow of the permissive LT movement as a function of the opposing volume

are designed to minimize the total intersection delay. The procedure applied here generates settings very similar to minimum delay settings. It is also computationally attractive.

The concept of the procedure consists of assigning green time for each turning movement according to its flow-to-capacity ratio. More precisely, for each phase a critical movement is considered, that is one which requires the longest green time. The procedure consists of a sequence of equations, which in general form can be defined as follows:

$$C = z(\mathbf{y}) \tag{15}$$

$$g_p = u_p(\mathbf{y}, C), \quad p \in P \tag{16}$$

where C = cycle length
$\mathbf{y}$ = vector of volume-to-saturation flow ratios y^i, $y^i = f^i/S^i$ for all movements i at the intersection
g_p = time assigned for phase p
P = set of all phases

In the case of a four-leg intersection like the one in Figure 2 and a two phase pattern, relationships (15) and (16) take the following form:

$$C = \frac{A}{1 - y^{NS} - y^{EW}} \tag{17}$$

$$g^i = (C - b)\frac{y^i}{y^{NS} + y^{EW}}, \quad i = \{NS, EW\} \tag{18}$$

where A and b are parameters.

Because of the mutual dependency between the saturation flow procedure and signal setting procedure, both need to be recomputed iteratively 3 to 4 times to obtain consistent results. An example of the relationship between the green time for one approach and the volume on the cross street at a four-leg intersection is shown in Figure 3.

Delay The delay function is a relationship between the flow and the delay experienced by the vehicle at the intersection. For a signalized intersection, the delay d can be expressed in general as follows:

$$\mathbf{d} = \mathbf{d}(\mathbf{f}, \mathbf{S}, \mathbf{g}) \tag{19}$$

The specific delay function applied in the study has the following form:

$$d_a = \frac{0.5C(1-u)^2}{1-ux} + 900T\gamma\left[\ x - 1 + \sqrt{(x-1)^2 + \frac{8(x-0.5)}{KT}}\ \right] \tag{20}$$

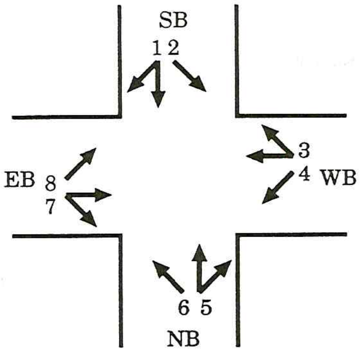


Figure 2: Four leg intersection used in the example

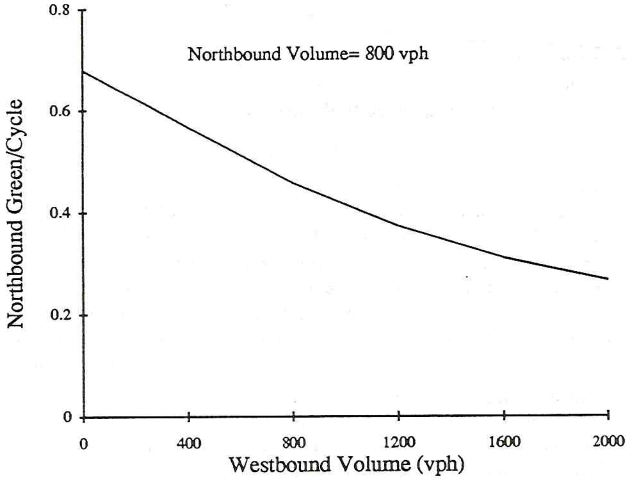


Figure 3: Green split for one approach as a function of the volume on the cross street

where d_a = average delay per vehicle for link a (sec/vehicle)
C = signal cycle length (sec)
$u = g/C$ = green split
g = green time (sec)
$x = f/K$ = volume-to-capacity ratio
T = duration of the flow (hrs)
γ = 1 for $x > 0.5$ and 0 otherwise

Capacity K is determined from the following relationship:

$$K = \frac{g}{C}S \tag{21}$$

where S is the saturation flow. Figure 4 shows an example of function (20).

The first term, called *uniform delay*, was originally developed by Webster (1958). It reflects the average delay experienced by drivers in the *undersaturation conditions*, that is when the arrival flow does not exceed capacity. In oversaturation conditions, $x = 1$ is used in the uniform delay term. The second term of function (20) is called *overflow delay*. It reflects the delay experienced by the vehicles when the flow rate is close to the capacity or exceeds it. Temporary overflow of the intersection occurs also when the average arrival rate is lower than the capacity, due to a random character of the arrival pattern. The earliest delay functions (for example, Webster, 1958) were based on the steady-state model

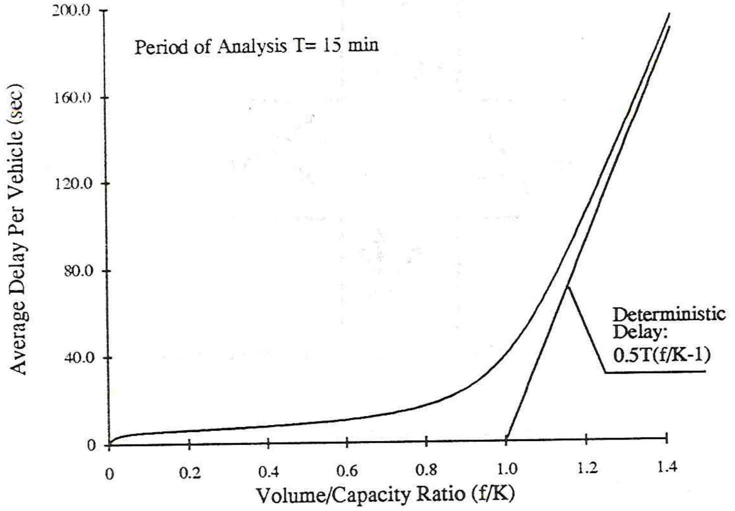


Figure 4: Steady-State Delay Model vs. Time Dependant Formula

and were defined only for undersaturation conditions when the volume-to-capacity ratio $x < 1$. Functions of this kind are not suitable to be application within a route choice problem, even for an undersaturated network if an oversaturation condition can occur during the solution process. Several alternative functions have been proposed to overcome this limitation. Since the oversaturation condition is best described by *deterministic queuing model*, an effort was made to reconcile the steady-state model for undersaturated conditions with the deterministic model for the oversaturation regime, especially for the intermediate condition when flow is close to capacity. Since these two models are based on very different assumptions, the task is rather delicate. The above function is one of a family of functions reconciling both conditions.

3.1.3 Unsignalized Intersections

Delay models for unsignalized intersections, whether major/minor priority intersections or all-way-stop controlled intersections, are much simpler than models for signalized intersections. In all cases, first the capacity is estimated, followed by the delay based on the flow-to-capacity ratio. The delay functions are similar the second term of delay functions for signalized intersection. Only a few notes are given here concerning the models for unsignalized intersections.

For priority intersections the 1985 HCM capacity model is applied. The HCM capacity procedure is computationally attractive and generates rather realistic results in a wide range of traffic conditions. For a comparison of alternative capacity models, see Troutbeck (1991). For delay estimation, the time-dependent formula developed by Kimber and Hollis (1979) is chosen.

The capacity of all-way-stop controlled intersections is modeled using the analytical iterative procedure developed by Richardson (1987). An analytical procedure is preferred over a regression-based one because of the wide range of traffic patterns occurring during the equilibration process. Regression-based models reported in literature (for example, Kyte, 1991), are validated only within a narrow range of traffic flows. A time-dependent delay function was developed using the technique by Kimber and Hollis (1979).

3.1.4 Freeway Related Facilities

Several kinds of freeway-related facilities occur within the test area: basic freeway segments, ramps and ramp-freeway junctions, weaving sections and toll plazas. All of them are modeled using appropriate capacity models and time-dependent delay functions similar to the delay function for signalized intersections (only the second term). Most of the capacity models are based on 1985 HCM. Some of them, however, had to be developed for oversaturation conditions, which cannot be avoided during the equilibration process of solving a UO route choice problem.

4 Test Area And Network

The study area depicted in Figure 5 is located in northwest suburbs of Chicago and covers about 300 square miles (800 square km). To develop the test network, detailed intersection control, layout and

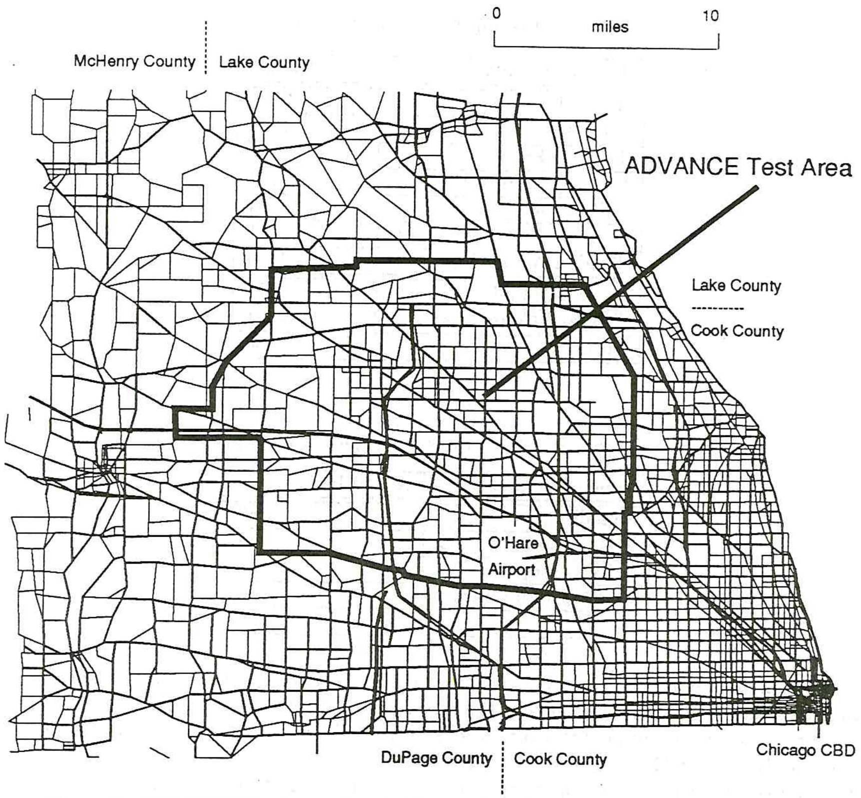


Figure 5: ADVANCE test area located in north and nortwest suburbs of Chicago

Table 1: Size of the study network

Number of Links	Number of Nodes	Number of Zones
7850	2552	447

some traffic data were collected in the field and merged with topology data from a regional planning network created by the Chicago Area Transportation Study (CATS). The process of coding and testing the network was made possible by using the Network Display Tool (Ramakrishnan et al, 1994). The network was then extensively tested and debugged for coding errors. Tables 1, 2 and 3 show the size and various characteristics of the test network.

In a conventional route choice model, the network is defined so that each intersection is represented as a single node and each road segment is represented as a single approach link (precisely a two-directional segment is represented as two links, one in each direction). This network representation defines travel times in terms of approach. In this study turning movement travel times are considered. This requires an appropriate network representation which is described below.

The approach consists of defining a special network representation so that each turning movement is represented by a separate link called an ***intersection link.*** More precisely, an approach node is defined for each approach to the intersection, and the number of intersection links originating from this node equals the number of turning movements. Similarly, an exit node is defined for each exit from the intersection. For a typical four-leg intersection with two-way approaches without turning restrictions, four approach nodes, four exit nodes and twelve intersection links are required in this representation (see Figure 6).

Networks of this type were used in different variations for detailed intersection analysis e.g. by Fisk

Table 2: Number of Turning Movement Links by Facility Type

Type of Facility	No. of Links
Arterial	2715
Collector	1346
Tollway/Freeway	197
Freeway Ramp	202
Toll Plaza	14
Freeway Weaving Section	11
Total of Actual Links	4485
Approach Links	874
Centroid Connectors	2491
Total	7850

Table 3: Intersections by Type of Control

Type of Control	No. of Intersections
Signalized	699
Major/minor priority	821
All-way-stop	60
Freeway-ramp merge	99
Total	1679

(1978) or Florian and Nguyen (1976). The disadvantage of this approach is that the number of links and nodes in a detailed network is much higher than in a conventional network. Also, to create a detailed network representation, a special procedure is required. This kind of network builder is described by Sharaf-Eldien (1988) and Meneguzzer et al (1990).

During the computations some operations need to be performed on an intersection-by-intersection basis; therefore the detailed network representation needs to retain the original intersection numbers either explicitly or by means of appropriately defined link and node labels. In this study the latter method is adopted from Meneguzzer (1990, p. 50). The disadvantage of this method is that for a network with say 5-digit node labels, the node labels in the detailed network have 10 digits, while the number of the nodes require only 5 or 6 digits. The program that creates this expanded network from the conventional network, called a *network builder*, also performs other functions such as:

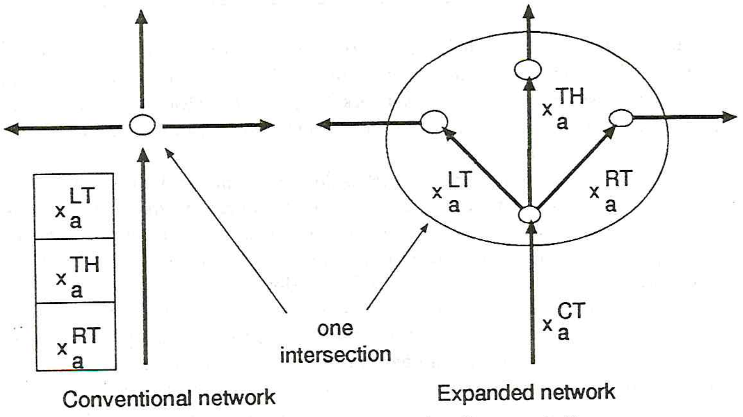


Figure 6: Expanded Intersection Representation

Table 4: Total Number of Trips per Hour

	Period	Time-of-Day	Internal Trips	Trips Total
1	Night	12pm-6am	7,839	19,439
2	AM Peak	6am-9am	83,708	184,185
3	Midday	9am-4pm	88,543	170,573
4	PM Peak	4pm-6pm	98,532	203,278
5	Evening	6pm-12pm	81,245	146,092

1. create dummy intersection approaches and exits so that all intersection approaches becomes two-way, which simplifies the network expansion procedure;

2. analyze actual intersection approach geometry and assign to the approach the closest intersection category defined before;

3. for each centroid connector attached to an intersection in the original network, create appropriate centroid connectors for the expanded intersection representation.

5 Travel Demand

Trip demand data for this study is based on CATS estimates for 1990. The process of creating trip tables from the CATS base data is described in Boyce et al (1994). There are two steps in this process. The first step consists of time-of-day factorization of the original 24-hour estimates. The factors are based on travel surveys performed by CATS tabulated to yield departure rates by 30-minute intervals. According to these data, five time-of-day periods are determined such that, within each period, the travel demand fluctuates as little as possible. The time-of-day periods chosen are shown in Table 4. The factors obtained in this way are applied to generate five time time-of-day matrices.

The second step consists of the analysis of the trips beginning or ending or both outside the test area. First, the conventional UO route choice problem is solved for the Chicago region which includes the test area. Routes generated are analyzed and *external zones* are created on the boundary of the test area to represent flows entering and exiting the test area. Each of the final matrices consists of the internal matrix and external trip matrix (see Table 4).

6 Model Solution

The implementation of the model is based on Meneguzzer (1990). Figure 7 shows the simplified structure of the algorithm. The algorithm is also described briefly in Section 2.

The algorithm starts with computation of initial estimates of link flows by turning movement. For these flows, capacities and other input parameters of delay functions are computed. Because of the mutual dependencies between the different parameters described in the section on delay functions, obtaining consistent values of these parameters requires iterative recomputation of the intersection analysis procedure not shown the figure. Capacities and other parameters of the delay functions obtained in this way are used to solve the standard UO route choice model. The resulting link flows are then used to reevaluate intersection capacities. The algorithm continues until the convergence condition is satisfied.

7 Computational Results

This section presents results of the asymmetric UO route choice model. The model has been implemented in Fortran 77. The computations were performed on a Sun workstation. As each Frank-Wolfe iteration required about 30-40 minutes of real time, one diagonalization iteration consisting of 6 Frank-Wolfe iterations required about 3-4 hours. The computational time for the intersection analysis is a small percentage (about 1%) of the total computational time.

A general observation is that the model converges faster or slower depending on the level of congestion; the more congested the network, the slower is the convergence. Results were evaluated using several global performance measures as well as by checking the reasonableness of the routes generated in the last diagonalization iteration. Table 5 presents the results obtained for the five time-of-day periods.

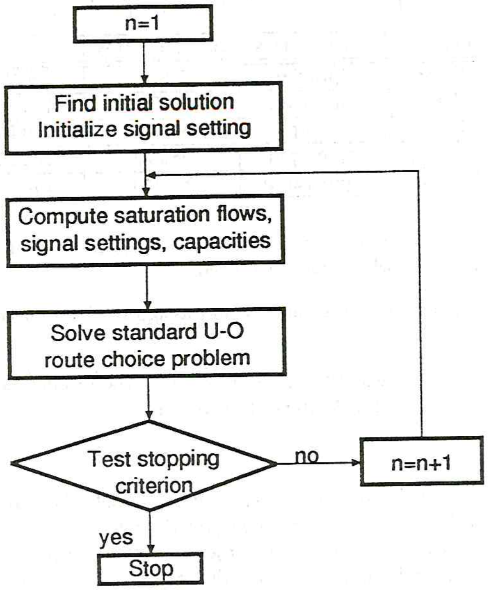


Figure 7: Implementation of Delay Models within an Asymmetric Route Choice Algorithm

Table 5: Network Performance Measures for the Asymmetric UO Route Choice Problem

	Time Period	Speed (mph)	v/c Ratio (-)	Travel Time (minutes)	Distance (miles)	Diagonal. Iterations	Links Converged (%)
1	Night	41.8	0.20	14.0	9.8	5	96.6
2	AM Peak	22.5	0.82	28.6	10.8	28	98.8
3	Midday	26.3	0.75	22.8	10.0	25	88.7
4	PM Peak	20.6	0.87	31.3	10.7	23	88.1
5	Evening	30.1	0.66	19.0	9.5	14	82.2

The speed shown in the table is the space-mean speed defined as ratio of total distance to total travel time; time is the average travel time, distance is the average trip distance for trips within the test area. The value of the convergence criterion is shown in the last column. The convergence criterion is the percent of the links in the network which converge. A link is considered to converge if its flow differs by no more than 10% from its flow in the previous diagonalization iteration. Within each diagonalization iteration, 6 Frank-Wolfe iterations are performed. Because of time constraints, only the first and second period problem were solved to convergence, which was set at the level of 95% of all links.

Another measure of network performance is presented in Table 6. Speeds in this table are space-mean speeds based on the last diagonalization iteration for each time-of-day period, computed separately for three functional classes. According to the table, the speeds for all road classes are 'inversely proportional' (not linearly) to the congestion level.

Figure 8 presents a sample of routes generated by the model in the last diagonalization iteration for one origin-destination pair. The thick outer line represents the boundary of the study area. Two rectangles represent the origin (thinner line) and destination (heavier line) zones. The six routes use the same links to a large extend, so that in the figure they overlap. Since delays are by turning movements, the routes do not include too many turns, which is frequently the characteristic of the routes generated by conventional route choice models.

Table 6: Travel Speed - Breakdown by the Road Class (mph)

	Time Period	Collectors	Arterials	Freeways
1	Night	36.3	39.0	62.6
2	AM Peak	16.9	20.0	41.7
3	Midday	20.8	23.8	45.8
4	PM Peak	15.7	18.0	39.5
5	Evening	24.3	27.8	49.0

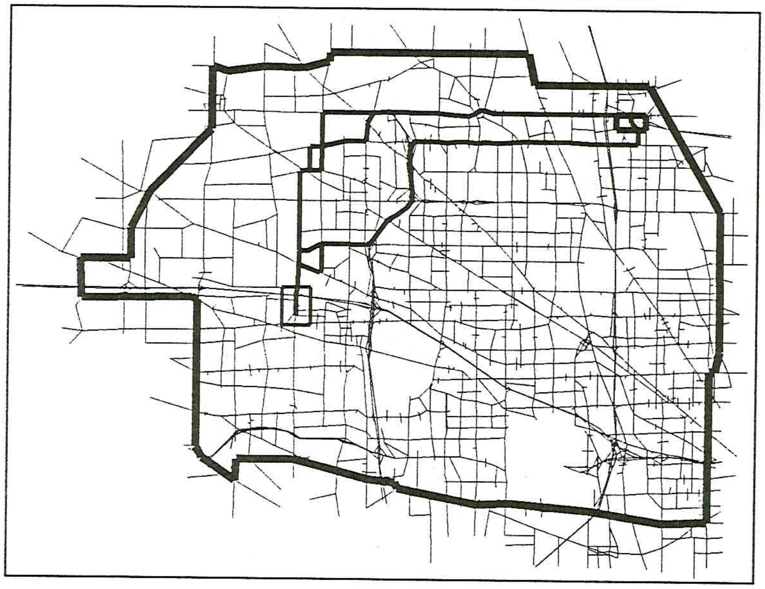


Figure 8: Routes generated by the model for one origin/destination pair

8 Conclusions

The purpose of the study was to generate link travel times in a transportation network for use as initial estimates for the ADVANCE, a dynamic route guidance system being implemented in Chicago suburbs. In general the estimates generated are satisfactory, even though certain links' travel times are unrealistic. Moreover, the estimates for different turning movements at the same intersection and some proximity of it, are consistent. Average travel speeds are similar to actual speeds according to traffic engineers familiar with the test area.

Two shortcomings of the model should be mentioned. First, in our model the intersection analysis is performed on an isolated intersection basis, so that the same flows are assigned for all intersection approaches along a route from an origin to a destination. In real world the flow passing through the intersection is never higher than its capacity and the next intersection analysis should reflect that. This shortcoming should be avoided and is a topic for future research. Treating intersections as isolated, morover, makes it impossible to account for progression effects along arterials leading to the delay overestimation. The third question is about the validity of the route choice behavior based on the travel time only. The road class (freeway vs. arterial vs. collector), which is not taken into account in our model, may be a significant factor of the route choice behavior as well.

Acknowledgements
The research was performed within the ADVANCE Project and was funded from this project. We would like to thank Dr. Andrew Tarko for his important contributions to this research especially in its traffic engineering aspect. Also, we want to acknowledge the National Center for Supercomputing Applications,

University of Illinois at Urbana-Champaign, for the computer resources provided for the research.

References

[1] Abdulaal, M., LeBlanc, L.J. (1979) Methods for combining modal split and equilibrium assignment models, *Transportation Science* **13**, 292-314.

[2] Akcelik, R. (1988) Capacity of a shared lane, *Australian Road Research Board Proceedings*, **14(2)**, 228-241.

[3] Beckmann, M.J., McGuire, C.B., Winsten, C.B. (1956) *Studies in the Economics of Transportation*, Yale University Press, New Haven, CN.

[4] Boyce, D.E., Chen, H.K., Rouphail, N.M., Sen, A. (1989a) *Subregional Route Choice Models with Link Travel Times Reflecting Intersection Flows*, Urban Transportation Center, University of Illinois, Chicago.

[5] Boyce, D.E., Kirson, A.M. and Schofer, J.L. (1990) *Scope, Feasibility and Cost of a Dynamic Route Guidance System Demonstration*, Prepared for the Illinois Department of Transportation by the University of Illinois at Chicago, Northwestern University, and Motorola, Inc.

[6] Boyce, D.E., Meneguzzer, C., Rouphail, N., Sen, A. and Lauritzen, T. (1989b) *A User-optimal Route Choice Model with Asymmetric Cost Functions Incorporating Intersection-related Travel Times*, Urban Transportation Center, University of Illinois, Chicago.

[7] Boyce, D.E., Zhang, Y., Hicks, J. (1994) *Trip Data Fusion in ADVANCE*, ADVANCE Working Paper, Urban Transportation Center, University of Illinois at Chicago.

[8] Dafermos, S. (1980) Traffic equilibrium and variational inequalities, *Transportation Science* **14**, 42-54.

[9] Dafermos, S. (1982) Relaxation algorithms for the general asymmetric traffic equilibrium problem, *Transportation Science* **16**, 231-240.

[10] Dafermos, S., Nagurney, A. (1984a) Sensitivity analysis for the asymmetric network equilibrium problem, *Mathematical Programming* **28**, 174-184.

[11] Dafermos, S., Nagurney, A. (1984b) Stability and sensitivity analysis for the general network equilibrium-travel choice model, *Proceedings of the Ninth International Symposium on Transportation and Traffic Theory*, Utrecht, The Netherlands, 217-231.

[12] Fisk, C. (1978) A transportation planning model for detailed intersection analysis, *Canadian Journal of Civil Engineering* **5**, 18-25.

[13] Fisk, C.S., Boyce, D.E. (1983) Alternative variational inequality formulations of the network equilibrium-travel choice problem, *Transportation Science* **17**, 454-463.

[14] Florian, M., Nguyen, S. (1976) An Application and Validation of Equilibrium Trip Assignment Methods, *Transportation Science* **10**, 374-390.

[15] Frank, M., Wolfe, P. (1956) An algorithm for quadratic programming, *Naval Research Logistics Quarterly* **3**, 95-110.

[16] Harker, P.T., Pang, J-S. (1990) Finite-dimensional variational inequality and nonlinear complimentarity problems: a survey of theory, algorithms and applications, *Mathematical Programming* **48**, 161-220.

[17] Hicks, J.E., Boyce D.E., Sen, A. (1992) *Static Network Equilibrium Models and Analyses for the Design of Dynamic Route Guidance Systems*, Urban Transportation Center, University of Illinois, Chicago.

[18] Improta, G., Candarelle, G.E. (1984) Control system design for an individual signalized junction, *Transportation Research* **18B**, 147-167.

[19] Kimber, R.M., Hollis, E.M. (1979) Traffic Queues and Delays at Road Junctions, *Transport and Road Research Laboratory Report* **909**, Crowthorne, Berkshire.

[20] Kyte, M. (1991) Interim materials on unsignalized intersection capacity, *Transportation Research Circular* **373**.

[21] Meneguzzer, C., Boyce, D.E., Rouphail, N., Sen, A. (1990) *Implementation and Evaluation of an Asymmetric Equilibrium Route Choice Model Incorporating Intersection-Related Travel Time*, Urban Transportation Center, University of Illinois, Chicago.

[22] Nagurney, A. (1984) Comparative tests of multimodal traffic equilibrium methods, *Transportation Research* **18B**, 469-485.

[23] Patriksson, M. (1991) *Algorithms for Urban Traffic Network Equilibria*, Department of Mathematics, Linkoping University, Linkoping, Sweden.

[24] Ramakrishnan, R., Berka, S., Hicks, J. (1994) *Network Display Tool: An Object Oriented Visualization Method for Intelligent Vehicle-Highway Systems*, Urban Transportation Center, University of Illinois, Chicago.

[25] Richardson, A.J. (1987) A delay model for multiway stop-sign intersections, *Transportation Research Record* **1112**, 107-114.

[26] Rouphail, N.M., Radwan, A.E. (1989) Simultaneous optimization of signal settings and left turn treatments, paper presented at the 69th Annual Meeting of the Transportation Research Board, Washington D.C.

[27] Said, K., Rouphail, N.M., Boyce, D.E., Sen, A. (1993) *Optimization of Traffic Signal Controls within Equilibrium Route Choice Models*, Urban Transportation Center, University of Illinois, Chicago.

[28] Sharaf-Eldien, O.M. (1988) *Hierarchical Traffic Flow Optimization for Congested Networks*, Ph.D. thesis, University of Maryland.

[29] Smith, M.J. (1979) The existence, uniqueness and stability of traffic equilibria, *Transportation Research* **13B**, 295-304.

[30] Smith, M.J. (1982) The existence, junction interactions and monotonicity in traffic assignment, *Transportation Research* **16B**, 1-3.

[31] Stewart, G.W. (1973) *Introduction to Matrix Computation*, Chapter **3**, Academic Press, New York.

[32] Transportation Research Board (1985) *Highway Capacity Manual*, Special Report **209**, Washington D.C.

[33] Transportation Research Board (1993) *Basic Freeway Section*, Revised Chapter **3** of the 1985 Highway Capacity Manual.

[34] Troutbeck, R. (1991) *Recent Developments in the Theory of Unsignalized Intersections*, Queensland University of Technology, Brisbane.

[35] Wardrop, J.G. (1952) Some theoretical aspects of road traffic research, *Proceedings of the Institution of Civil Engineers*, Part **II**, 1, 325-378.

[36] Webster, F.V. (1958) Traffic signal settings, *Road Research Technical Paper* **39**, Her Majesty's Stationary Office, London.

[37] Webster, F.V., Cobbe, B.M. (1966) Traffic signals, *Road Research Laboratory Technical Paper* **56**, Her Majesty's Stationary Office, London.

METACOR : A macroscopic Modelling Tool for Urban Corridor

N. ELLOUMI, INRETS, France
H. HAJ-SALEM, INRETS, France
M. PAPAGEORGIOU, Technical University of Munich, Germany

ABSTRACT

The paper describes a new integrated macroscopic modelling tool for simulating traffic flow phenomena within urban corridors including both motorway and urban road subnetworks of arbitrary topology. The new modelling tool METACOR emerged from expansion and integration of two existing modelling tools METANET and SSMT for traffic flow on motorway networks and on urban networks respectively. In this paper, the basic equations including traffic assignment modelling of the two traffic models are described and the main characteristics of METACOR are outlined. METACOR has been validated on the basis of real traffic flow measurements selected under a broad spectrum of traffic conditions. The mathematical model is capable of describing complicated traffic phenomena with acceptable accuracy. A simulation program which is developed on the basis of the modelling equations, mays be used as a tool for testing of control strategies related to ramp metering , signal control and Variable Message Sign (VMS) control.

I. INTRODUCTION

Modelling of traffic flow on a corridor is a useful tool for several traffic engineering tasks such as:

- Development and evaluation of traffic control strategies (ramp metering, VMS strategies, urban intersection control, etc.)
- Short-term prediction and surveillance of traffic state in complex networks
- Evaluation of the impact of new constructions, comparison of alternatives etc.
- Evaluation of the impact of capacity reducing events (e. g. works, incident or accident) or increased demand etc.

A corridor may be defined in a wide sense as a mixed (road and motorway) network. In a weaker sense, a corridor is defined as a traffic system including a motorway stretch, a parallel urban arterial, and the connecting links between them. In the past, traffic control systems within corridors have been developed independently for each control measure attempting to optimize traffic flow on either the motorway or the urban road network separately. The developped model METACOR (Modèle d'Ecoulement du TrAfic sur CORidor) is a tool for testing integrated control approaches that take into account the overall traffic conditions on the motorway and the urban network. The interest in integration of control measures in general, and in corridor systems in particular, has been increasing in Europe, North America, and Japan over the last 5 years.

This paper presents the development of a theoretical modelling framework for traffic flow in multidestination, multiple route choice, mixed networks on a purely macroscopic basis by using two existing models namely METANET (Modèle d'Ecoulement du Trafic Autoroutier en NETwork), see Ref [1], [2], already developed in the frame of DRIVE 1 Programme V1035 CHRISTIANE and SSMT (Simulation Semi-Macroscopique du Trafic) see ref [3], [4]. In order to meet the model specifications and enable a unified approach, some essential modifications of SSMT with regard to route choice behaviour of subflows with different destinations have been introduced.

The macroscopic approach is important in that it leads to relatively low computational effort and eventually allows for simulation of traffic flow on large scale networks. On the other hand, the macroscopic approach requires adequate modelling of complicated route choice phenomena which are more easily treated in a microscopic scale for the price of excessive computational effort.

Organization of this paper is the following: Section II presents the development of a mathematical model, i.e. derivation of the mathematical model structure and equations. Section III describes the real data available and the parameter estimation. Constant parameters included in the model equations reflect particular characteristics of a given traffic system depending upon street geometry, vehicle characteristics, drivers' behaviour etc. Constant parameters should hence be specified so as to fit a representative set of real data with maximum accuracy. Section IV corresponds to the model validation. The simulation models are applied to traffic data different from the ones used for parameter estimation. In particular, accuracy of the model is investigated under several traffic conditions, times of the day, locations etc.

II. DESCRIPTION OF THE CORRIDOR TRAFFIC FLOW MODEL "METACOR"

II.1 Network representation and Input Data

The corridor network is represented by a directed graph. More precisely, urban signalized junctions, motorway bifurcations, junctions and on/off ramps are represented by the nodes of the graph whereas the motorway and the urban stretches between these locations are represented by the links. The two directions of a motorway or of an urban stretch are modelled as separated links with opposite directions. Inside each link we suppose homogeneous geometric characteristics such as number of lanes, upgrade, curvature, etc... An inhomogeneous motorway stretch may be represented by two or more consecutive links separated by nodes at the locations where the change of geometry occurs. At the bounds of the network, origin or destination links are added where traffic respectively enters or leaves the simulated part of the network.

The traffic volumes entering via the origin links - and, optionally, their mean speed - must be provided as input to the simulation. Moreover, the composition of entering traffic with respect to the different destinations in the network must also be provided. At the exits of the network, traffic flow may be influenced by the traffic conditions in downstream stretches (e.g. spillback of congestion). If available, traffic densities at the destinations may therefore also be used as input data.

II.2 Traffic variables

The simulation of traffic behaviour in the network, both for the motorway and the urban links, is based on a macroscopic modelling approach. The model's aggregate variables are the density ρ(veh/km/lane), the mean speed v(km/h) and the traffic volume (or flow) q(veh/h).

For a space/time discretized presentation, we define traffic density $\rho_{m,i}(k)$ as the number of cars in the section i of link m at time t=kT, divided by the section length L_m where T denotes the sample time interval and k = 0, 1, 2, ... is a discrete time index. Similarly we define mean speed $v_{m,i}(k)$ as the mean speed of the cars included in the section i of link m at time $t=kT$. Finally, traffic volume $q_{m,i}(k)$ is the number of cars leaving the section i of link m during $kT< t < (k+1).T$, divided by T .

II.3 Basic equations

As mentioned before, METACOR consists of the integration of two macroscopic models namely METANET and SSMT. Consequently two main mathematical equation sets constitute the traffic flow modelling process for the Corridor Network: the motorway part and the urban part.

II.3.1 Modelling of the Motorway links

For modelling purposes, each link is subdivided in segments or sections with typical lengths of 300 to 800 meters (Fig. 1). The above-mentioned traffic variables are calculated for every segment i by means of the following non-linear difference equations (see Ref [3,4] for details) :

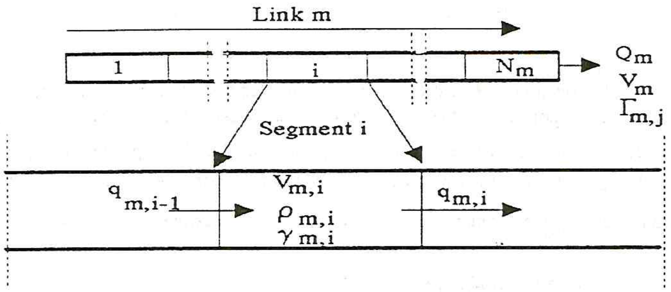


Fig. 1: Space discretization of the motorway link

Continuity Equation :

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + \frac{T}{L_m \lambda_m}\left[q_{m,i-1}(k) - q_{m,i}(k)\right] \tag{1}$$

Relationship between flow and speed :

$$q_{m,i}(k) = \rho_{m,i}(k).v_{m,i}(k).\lambda_m \tag{2}$$

Speed equation :

$$v_{m,i}(k+1) = v_{m,i}(k) + \frac{T}{\tau}\left[F(\rho_{m,i}(k)) - v_{m,i}(k)\right]$$

$$+ \frac{T}{L_m} v_{m,i}(k)\left[v_{m,i-1}(k) - v_{m,i}(k)\right]$$

$$- \frac{\nu T}{\tau L_m} \frac{\left[\rho_{m,i+1}(k) - \rho_{m,i}(k)\right]}{\left[\rho_{m,i}(k) + \kappa\right]}$$

on- ramp term

$$- \frac{\delta T}{L_m} \frac{\left[u_{m,i}(k).V_{m,i}(k)\right]}{\left[\rho_{m,i}(k) + \kappa\right]}$$

Lane drop term

$$- \frac{\phi T}{L_m}\left[\frac{\lambda_i - \lambda_{i+1}}{\lambda_i}\right] \frac{\rho_i(k)}{\rho_{cr,i}} V_i^2(k) \tag{3}$$

Fundamental diagram :

$$F(\rho_{m,i}(k)) = v_{f,m} \exp\left[-\frac{1}{a_m}\left(\frac{\rho_{m,i}(k)}{\rho_{crm,i}}\right)^{a_m}\right] \tag{4}$$

τ, ν, κ, δ, ϕ are constant parameters which are given the same values for all network links. λ_m denotes the number of lanes in the link m whereas L_m is the length of each segment of the same link; the parameters $v_{f,m}$ (free-flow speed), $\rho_{cr,m}$ (critical density) and the exponent a_m are specific for the fundamental diagram of the corresponding stretch (link) m. Different input options for specification of the fundamental diagram are offered by the program. T is the simulation time step and k denotes the time step presently in calculation ($t=kT$).

In order to be able to consider different destinations, some composition rates have been introduced for each segment. They represent portions of the traffic volume $q_{m,i}$ destinated to corresponding destination links. If J_m is the set of destinations that are reachable through link m, $|J_m\text{-}1|$ independent composition rates $\gamma_{m,ij}$ are required for each segment i of link m where j is the index of the corresponding destinations. The modelling of the composition rates $\gamma_{m,ij}$ in segment i is done by definition of destination oriented partial densities $\rho_{m,ij}$:

$$\rho_{m,ij}(k) = \rho_{m,i}(k)\cdot\gamma_{m,ij}(k) \tag{5}$$

$$\rho_{m,i,j}(k+1) = \rho_{m,i,j}(k) + \frac{T}{L_m\lambda_m}\left[\gamma_{m,i-1,j}(k)q_{m,i-1}(k) - \gamma_{m,i,j}(k)q_{m,i}(k)\right] \tag{6}$$

For every time step k the traffic variables are calculated according to Equations (1) - (6) for all segments in all links. For simplicity of notation, we denote by Γ_{mj} and Q_m the composition rate and the traffic volume of the last segment of the link m

II.3.2 Modelling of the Urban links

The main dynamic traffic phenomena to be described within urban links are not due to the inherent traffic flow dynamics (like in motorway links) but due to the switchings of the traffic lights at urban junctions and the corresponding shock waves generated by the stops of cars in front of the red lights. Therefore, a simple linear relationship may be chosen for description of mean speed within urban links (see ref [5,6,8]). Thus time evolution of the three traffic variables in a section of a given link may be given by:

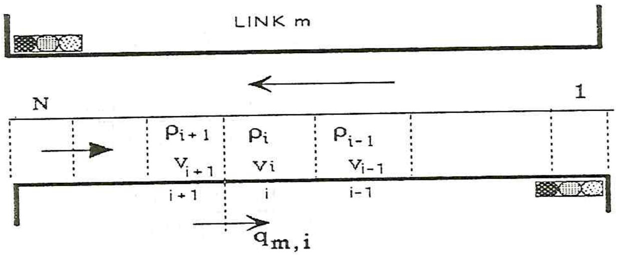


Fig. 2: Segmentation of the urban links

$$\rho_{m,i}(k+1) = \rho_{m,i}(k) + T / L_{m,i}\,[q_{m,i}(k) - q_{m,i-1}(k)] \tag{7}$$

$$q_{m,i}(k) = \rho_{m,i}(k)\,v_{m,i}(k) \tag{8}$$

$$v(\rho_{m,i}(k)) = v_f\,(1 - \rho_{m,i}(k) / \rho_{max}) \tag{9}$$

where

v_f : free speed

ρ_{max} : critical density (all lanes),

T : Sample time interval for the simulation. For discretisation stability reasons, T should be chosen such that $T \le MIN(L_i/v_f)$. This means that in the same time interval one vehicle can not move from the section j to j-2.

Equations (9) may be replaced into (8) to give $q_{m,i}(k) = Q[\rho_{m,i}(k)]$ and this may be substituted into Eqn. (7), the latter being the main model equation. It can be shown however, that this model discretization is numerically unstable in case of congestion within the link. For this reason, the following treatment is considered in the urban link (SSMT model basis):

> In order to take into account the effect of a downstream congestion, the traffic flow is calculated according to the downstream and upstream section densities. This means that we must define the shockwave direction at each point X1 and X2 (Fig.3) corresponding to the space discretisation. So, we have three different cases for calculating the traffic flow leaving the section i and entering the section i-1, according to the comparative density values in the sections i and i-1:

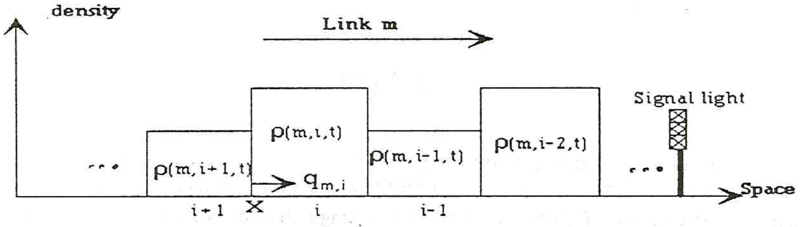


Fig. 3: Space/time discretization of SSMT model

We denote around the discontinuity point X by $Q_{max,i+1}$, $Q_{max,i}$,q_{dwn} , q_{ups} the capacity of the segment i+1, the capacity of the segment i, the admissible traffic volume at the downstream within segment i+1 and the admissible traffic volume within segment i respectively. The calculation of the traffic flow $q_{m,i}$ is effectuated as follow:

$$q_{ups}(k) = \begin{cases} Q[\rho_{m,i+1}(k)] & \text{if } \rho_{m,i+1}(k) \le \rho_{cr,i+1} \quad (\rho_{cr,i+1} : \text{critical density} = \rho_{max,i+1}/2) \\ Q_{max,i+1} & \text{if } \rho_{m,i+1}(k) > \rho_{cr,i+1} \end{cases}$$

$$q_{dwn}(k) = \begin{cases} Q_{max,i+1} & \text{if } \rho_{m,i}(k) \le \rho_{cr,i} \quad (\rho_{cr,i} : \text{critical density} = \rho_{max,i}/2) \\ Q[\rho_{m,i}(k)] & \text{if } \rho_{m,i}(k) > \rho_{cr,i} \end{cases}$$

Finally $q_{m,i}$ is calculated as :

$$q_{m,i}(k) = \text{MIN}[q_{ups}(k)\, ,\, q_{dwn}(k)] \tag{10}$$

This calculation allows for existence of different capacities at successive sections, e.g lane drop or augmentation of the number of lane. After the traffic flow calculation, the traffic density in each section i is provided from the incoming and outcoming traffic flow of the corresponding section by using equation (7).

For the subflow by destination j, the same approach as in motorway links (METANET) is used. Some composition rates have been introduced for each segment according to the eqn. (5).

According to the traffic lights state, the traffic flow exiting a link is updated either taken as calculated according to the above description (in case of green signal) or is set equal to zero (in case of amber or red signal). Moreover, a starting phase of a few seconds with reduced flow values is taken into account at the beginning of green.

II.3.3 Modelling of the Motorway Nodes

Motorway bifurcations and junctions (including on- and off-ramps) are, as already mentioned, represented by the nodes of the model (Ref [1]). Generally speaking, traffic enters a node *n* through a number of input links (members of the set I_n) and is distributed to a number of output links (members of the set O_n). For proper calculation of the distribution, the destinations of the inflows to the node must be considered. The partial flow with a certain destination *j* in a leaving link is calculated according to the total incoming traffic destinated to *j* and according to the percentage of users who choose the corresponding route. Expressed in equations, the relations for each node *n* read:

$$q_{nj} = \sum_{m \in I_n} Q_m \Gamma_{mj} \qquad \forall j \in J_n \text{ where } J_n = \bigcup_{m \in I_n} J_m \tag{11}$$

$$q_{m,0} = \sum_{j \in J_m} q_{nj} . \beta_{nj}^{m} \qquad \forall m \in O_n \tag{12}$$

$$\gamma_{m,o,j} = \beta_{nj}^{m} \frac{q_{nj}}{q_{m,o}} \qquad \forall m \in O_n \ \forall j \in J_m \tag{13}$$

Equation (11) expresses the merging of the inflows in the node, q_{nj} being an auxiliary variable which represents the total flow entering node *n* and destinated to *j*. Equation (12) distributes the incoming flows to the leaving links, and Equation (13) provides the corresponding initial composition rates $\gamma_{m,oj}$. The splitting rate β_{nj}^{m} determines the fraction of q_{nj} which leaves through link *m*. For example, if β_{nj}^{m} equals 1, all drivers at node n take a route via link m to their destination *j*. On the other hand, if all drivers take other alternative routes to destination *j*, β_{nj}^{m} is zero. Consequently the splitting rates of the corridor network, organised in a vector β, express the average route choice behaviour of the drivers at the network nodes.

II.3.4 Modelling of the Urban Junction

The modelling of the urban intersection is more complicated than the motorway node (see Ref [8]). For the motorway node, all node equations are static. In the contrary, for the urban intersection all traffic equations are dynamic. Generally speaking, the mathematical equations used for the intersection modelling are similar as for link modelling. However, different driver behaviour modelling is used, especially concerning the left turn modelling so as to reflect a realistic representation of real phenomena. This part of the model was taken from SSMT model.

As indicated in Fig. 4, the intersection can have from one to four entering links and one to four exiting links. In order to describe the interaction between a given movement and another, the inner junction is divided into spatial zones corresponding to each movement. These zones play the same role as the sections for a link. We note that the total number of movements in a junction (excluding U-turns) may be up to 12 (three movement per entering link).

The entering traffic flow for each movement is calculated by the same formula as for a link considering the corresponding upstream last section of the link and the density in each corresponding intersection zone. Fixed or variable turning rates may be defined for each movement in the junction. The exiting traffic flow is provided in dependence of the traffic variables within the corresponding zone in the junction and in the first section of the downstream link. In this way, queue spillback from the downstream link can be taken into account.

Concerning the left-turning vehicles, the treatment is a little more elaborated. The exiting traffic flow depends also on the traffic flow of the priority lane (opposite direction). The left-turning delay is described according to the driver impatience theory. The principle of the theory is to leave a vehicle pass through the junction if the gap between two successive vehicles of the priority lane is sufficiently long. This gap length value decreases as the time elapses.

In SSMT junction model, the multiorigin, multidestination traffic modelling is not included. Similarities between the splitting rates in METANET and the turning movement rates in the intersection exist as indicated in the equation (6) which is used for the two model link approaches. Corresponding improvements and extensions of subflow modelling with different destinations at the intersection and the node have been introduced.

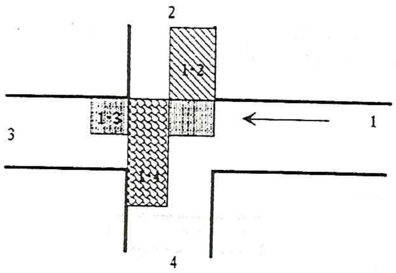


Fig. 4 : Urban intersection modelling

Since the subflows with different destinations and their respective route choice are part of the problem formulation, our approach is to introduce the correponding variables in SSMT model without loosing previous variables like turning mouvements. On the basis of this formulation of node equations, and in order to conserve the turning movements used in the original SSMT equations, we must calculate the turning movements in terms of the subflows, the splitting rates, and the composition rates. We note that, turning rates in SSMT are given as traffic independent parameters which in METACOR become internal traffic flow variables depending upon the drivers route choices and the traffic composition by destination.

For the internal calculation of respectively the subflow per destination and the turning movement rates, the following steps have been adopted:

Consider an urban junction N among the several intersections of the urban network. Assume S_m the set of destinations reachable from the link m and $S_{m'}$ the set of destinations reachable throught the link m'. These links m, m' are included in the four branches of the urban junction. The internal subflow calculation at each link gives the $\Gamma^d{}_m$ which correspond to the fraction of flow destinated to the set of possible destinations. The traffic flow entering the junction by link m and leaving by link m' is destinated to a set of destination $S_{m,m'} = S_m \cap S_{m'}$. Hence, the internal turning rate *α(m,m')* calculation is computed as:

$$\alpha_N^{m,m'} = \sum_{d \in S_m \cap S_{m'}} \Gamma_m^d \beta_{N,d}^{m'} \tag{14}$$

Equation (14) permits the use of the intersection modelling as it is included in SSMT. Now, the problem is the following: How to calculate the input subflow by destination at each output link m'of the junction i.e. the composition rate $\gamma^d{}_{m'}$ for the first segment of the output links of the junction (see fig 5.) ?

As indicated before, the intersection modelling in SSMT is similar than the link modelling. This means, that the calculation of input/output flows into the junction considers the length and the capacity of the intersection and the conservation equation is applied to each movement and to the intersection as a whole. In this way, the total input and output flows can be calculated by using intersection modelling. Assume $Q_{N,in}$, $Q_{N,out}$, *Nvh*, Nvh^d the total calculated input/ output flows, the total number of the vehicles present, the number of vehicles destinated to the destination *d* in the Junction *N* respectively (see Ref [9]). The last two variables (*Nvh* ,Nvh^d) are considered now as dynamic variables. The conservation equations for the number of vehicles present in the intersection *N* are given as :

$$\mathrm{Nvh}\,(k+1) = \mathrm{Nvh}\,(k) + T\,(\,Q_{N,in}\,(k) - Q_{N,out}\,(k)) \tag{15}$$

$$\mathrm{Nvh}^d\,(k+1) = \mathrm{Nvh}^d\,(k) + T\,(\,q^d{}_{N,in}\,(k) - \gamma^d{}_{tmp}(k) Q_{N,out}\,(k)) \tag{16}$$

where by definition $$q_{N,in}^d = \sum_{m \in I_n} \Gamma_m^d Q_{N,in} \tag{17}$$

I_n represents the set of the input links of the intersection.

γ^{d}_{tmp} (k) = an intermediate composition rate at the intersection of the subflow destinated to destination *d*.

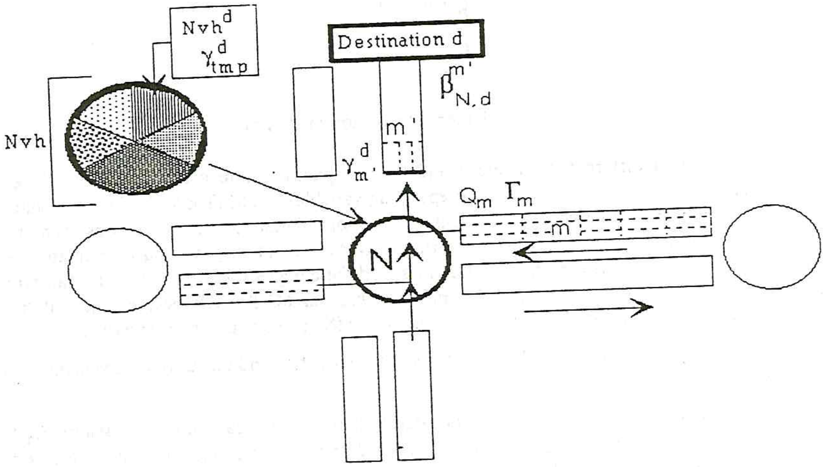


Fig. 5. Sub-flow problem configuration

By definition :

$$\gamma^{d}_{tmp}(k) = Nvh^{d}(k) / Nvh(k) \tag{18}$$

The final composition rate $\gamma^{d}_{m'}$ of the link m' destinated to the destination d is calculated as:

$$\gamma^{d}_{m'}(k) = \frac{\beta^{m'}_{d}\gamma^{d}_{tmp}(k)Q_{N,out}(k)}{\sum_{d\in S_{m'}}\beta^{m'}_{d}\gamma^{d}_{tmp}(k)Q_{N,out}(k)} \tag{19}$$

Where $S_{m'}$ is the set of destinations reachable through the link m'. The equation (19) corresponds to the final step for the calculation of the subflow of each first segment of the output links of the junction.

III. METACOR APPLICATION TO THE CORRIDOR PERIPHERIQUE IN PARIS

Application of METACOR model to the southern part of the Corridor Périphérique (C.P.) is based on a subdivision of the motorway and the urban part into several links and sections. As far as the motorway is concerned, the strech is subdivided into 14 links with one section each. Concerning the urban part, the number of sections is calculated automatically according to the time interval chosen for the urban links and the intersection. Three time intervals have been chosen for the application of METACOR model. In general, the choice of each time interval T should be chosen in order to guarantee model stability according to $T < L/v_f$, where L = length of the link, v_f = free speed.

1. For the Motorway part:

In our case, the smallest section has a length of 400 m and the free speed is expected in the order of 90 km/h which gives T < 16 s. Thus T is chosen to be 12 s. Since the measurement sets are discretized on the basis of 6 min, the same measurement values are used for 30 successive time intervals.

2. For the urban part two different time intervals have been chosen:

For the link:The maximum speed in the urban area is expected in the order of 60 km/h. The smallest section has a length of 70 m, consequently the time interval Tu must be Tu < 5 second. Tu is chosen to be 4 seconds.

For the Junction:The time interval of the urban junction is chosen to be 1 second. This value garantees METACOR stability.

In summary, METACOR model is running with three time intervals, 12 , 4 and 1 second for respectively the motorway, the urban link and the urban intersection.

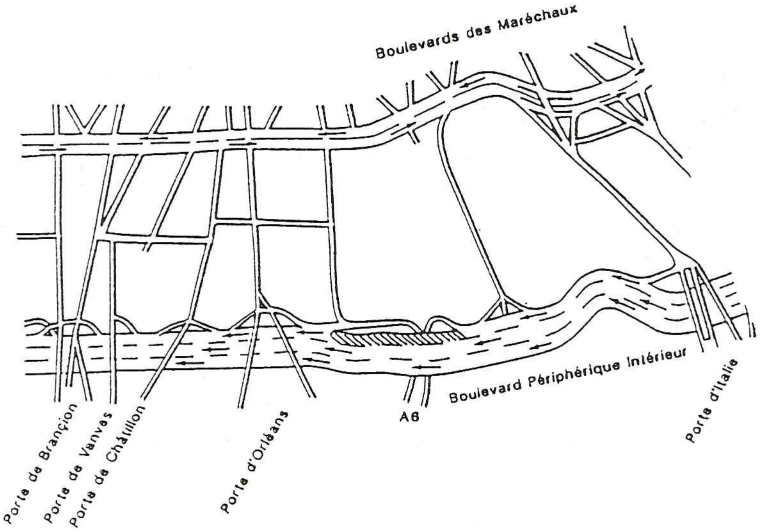
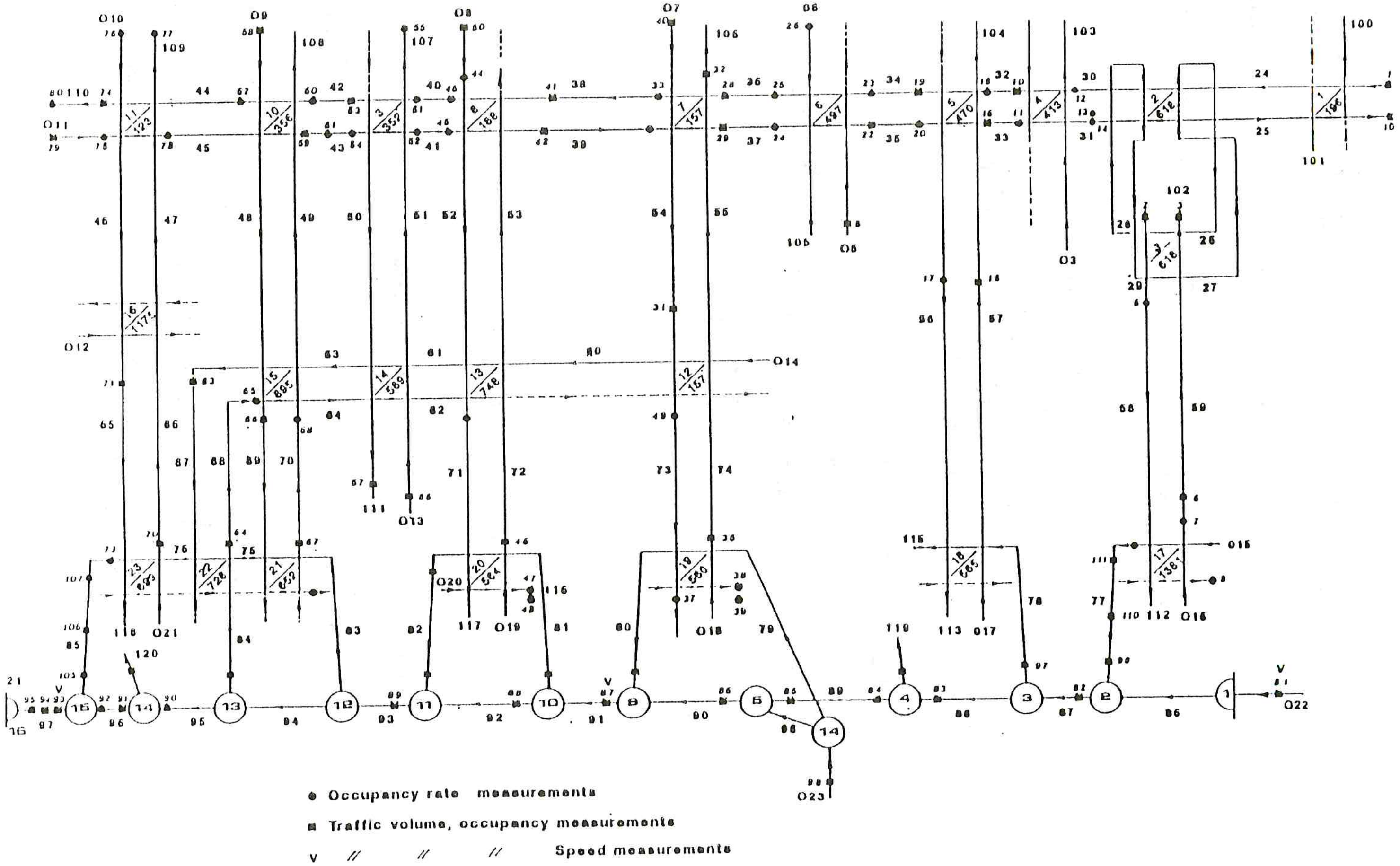


Figure 6. Simulated part of the Corridor Périphérique

III.1 Description of the Southern Part of the Corridor Périphérique

Figure 6 and 7 correspond to the simulated part of the Corridor Périphérique. Some simplifications of the topology of the urban area have been introduced. We consider only the main intersections where the traffic is congested during the rush hours (morning and evening).

The considered part of the Corridor Périphérique is saturated almost all day long due to the excessive on-ramp volume from the autoroute A6 entrance at Porte d'Orleans, and the reduction of the lane number (from 3 to 2 lanes) at the off-ramp Gentilly. Hence, the traffic conditions on this part are very difficult, whatever the time of the day. Every day, these congestions spread until Porte de Bagnolet at the peak hours (5 kilometres upstream of Porte d'Italie). Comparison of the off-ramp traffic flow in both congested and dense situations, shows that the average traffic flows at the off-ramps are higher at congested state. This is probably due to a diversion of the drivers, when severe congestion is present. On the other hand, the exit volumes of the different sections differ in a significant way.


O10
O9
O8
O7
O6
O5
O3
O11
O12
O13
O14
O15
O16
O17
O18
O19
O20
O21
O22
O23
100
101
102
103
104
105
106
107
108
109
110
111
112
113
115
116
117
118
119
120
● Occupancy rate measurements
■ Traffic volume, occupancy measurements
V // // // Speed measurements

Concerning the parallel urban arterial, similar phenomena as for the ringway of Paris are observed especially concerning the traffic conditions and the congestion propagation. The traffic problems on this urban arterial are more complex than in the ringway, due to the high traffic volume, coming from Paris City to the direction of the Paris surroundings and vice-versa, generating a high level of congestion on the secondary axis at the urban intersections.

III.2 Available Traffic Data

Parameter estimation and validation of the macroscopic simulation model METACOR requires availability of adequate real traffic data. This section presents the data utilized in the present study along with preliminary data processing and analysis.

Traffic data utilized were collected from the described part of the Corridor Périphérique. As shown in Fig. 6, the black spots indicate the location of available detectors in the considered network. As far as the motorway is concerned, all detector mesasurements correspond to traffic volumes, occupancy rates and speeds. Speeds are measured at three stations located respectively at the boundaries and in the middle of the motorway stretch.

On the other hand, the urban area is equipped with loop detecors which collect occupancy rates and the traffic volumes. The number of occupancy measurement stations is higher then the number of traffic volume stations. This is due to the real time algorithm used for the intersection control that is mostly based on occupancy rate measurements. Nevertheless, traffic volume measurement stations are located at all complicated intersections such as Italie, Orléans, Chatillon which represent the main traffic problem spots. However, in contrast to the motorway stretch, at the boundaries of the urban area several input / exit link intersections are not sufficiently equipped. This means that, for the urban area some data processing had to be used for the estimation of traffic volumes at the input/output intersections that are located at the boundaries of the considered network site.

For METACOR parameters calibration and validation, one week of real data has been used. A first step was parameters calibration using data of one day (05/11/92). The second step was the model validation using the same parameters found for the calibration day.

III.3 Signal plans during the Calibration and Validation of METACOR

The signal plans used as input data for this study were provided by the Ville de Paris site owners. These signal plans correspond to fixed plans with two cycle durations of 60 and 80 seconds. In real time, the implemented signal plan control strategy can choose between these two signal plans according to the traffic conditions on the urban part.

In the urban modelling part the signal states are updated each second corresponding to the time slice of the junction. Each urban link has a traffic light at its end. In the present version of METACOR, the traffic signals are pre-fixed. These traffic signals are described by their green, amber and red durations and by a phase number in order to describe in which group they belong in the junction. Finally, an offset describes their beginning of green inside a given cycle.

Even though the traffic lights are currently fixed, it is possible to introduce very easily a routine which modifies the traffic lights at each given time period, which can be at each time step.

III.4 Calibration of METACOR Model Parameters

The estimation of the unknown parameters of the macroscopic model is a nontrivial task since system equations are highly nonlinear in both the parameters and the state variables. The most common approach for the identification of nonlinear systems is the least squares output error method which minimizes the discrepancy between the model and the real process output with respect to some quadratic output error function. Thus, the parameter estimation problem may be formulated as the minimization of least squares output error.

For parameter estimation and model validation, it is necessary to distinguish input variables from output variables in the measurement data. Input variables are the boundary variables of the motorway stretch as well as the input/ output urban link volumes. Generally speaking, the input variables correspond to the Origin/Destination traffic volumes. The model being fed with these measured data, traffic volumes and mean speeds for the internal measurement sites, as produced by the model equations, may serve as output variables to be compared to the corresponding real traffic data.

The following nominal parameter set was established as a best compromise after several simulation runs on the measurement set of 11/05/92:

Motorway : $v_f = 90$ km/h , $\rho_{cr} = 37$ veh/km , a =2. τ =0,01 h , ν =35 km^2 /h , δ =0,8 and ϕ =2.

Urban : $v_f = 60$ km/h, $\rho_{max} = 180$ veh/km

Concerning the turning rates respectively at each node including off-ramp and urban junction which are considered as a model inputs, these values have been adjusted during some preliminary simulation runs.

IV. RESULTS

IV.1 Mean square deviation results

The qualitative (see output trajectories) and quantitve results obtained indicate that METACOR copes with the measurement sets in a satisfactory way. The congestions are reproduced in the same location and the same time as in real life on the urban and motorway part. However, due to the several detector failures at the internal links of the urban area, some outputs results were not very accurate. The tables I, II and III include the mean square deviations for respectively traffic volumes, speed of the motorway links and volumes of the urban links.

Data Sets	L88	L89	L90	L91	L92	L95	L96	L97
06/03/92	883	626	916	769	603	990	854	810
06/24/92	864	631	895	869	864	850	604	867

Table I. Volume mean square deviations (Motorway Part)

Data Sets	L88	L89	L90	L91	L92	L93	L96	L97
06/03/92	10,4	13,5	7,8	20,4	11,9	15,6	23,2	18,1
06/24/92	13	20	6	18,1	15,2	15,0	22,1	19,2

Table II. Speed mean square deviations (Motorway Part)

Data Sets	L33	L36	L37	L39	L42	L43	L110
06/03/92	990	703	938	1269	990	896	145
06/24/92	1008	1304	625	569	1300	970	150

Table III. Volume mean square deviations (Urban Part)

IV.2 Output Trajectories

The output trajectories, i.e. trajectories of the several internal stations for mean speeds and traffic volumes for two measurement sets, are depicted in the figures 8, 9, 10, 11, 12, and 13. As for all figures, solid lines depict measured trajectories whilst dashed lines depict the corresponding model outputs. Concerning the motoway part, the morning breakdown of mean speed observed in the measurement set (03/06/93) is reproduced fairly accurately by the model, see e.g. Fig. 8. Congestion mounting from donwstream (A6 on-ramp station number 85) and reaching station Italie (81) is propagated by the model in upstream direction with considerable accuracy as shown in figures 8 and 9. Of course stop-and-go traffic is less pronounced in the model trajectories which folow the real mean speed values in the average.

With respect to traffic volumes, figure 10 depicts a typical situation for measurement set 02/06/92, measurement stations 87 which correspond to the downstream of the on ramp motorway A6. Despite the standard deviation error of 650 veh/h, measured traffic volume is followed fairly accurately by the model trajectories both for fluid traffic and for congested traffic.

Finally, figures 11, 12, and 13 depicts the time evolution in the urban part with respect to traffic volume. The time evolution of the measured and simulated data follows the traffic conditions in the urban part. Thus, the urban modelling part reproduces at a sufficiently accurate level the measured traffic conditions. Althought, the estimation of turning movement rates has been a non trivial task, after several trials, these parameters have been estimated in relatively accurate way.

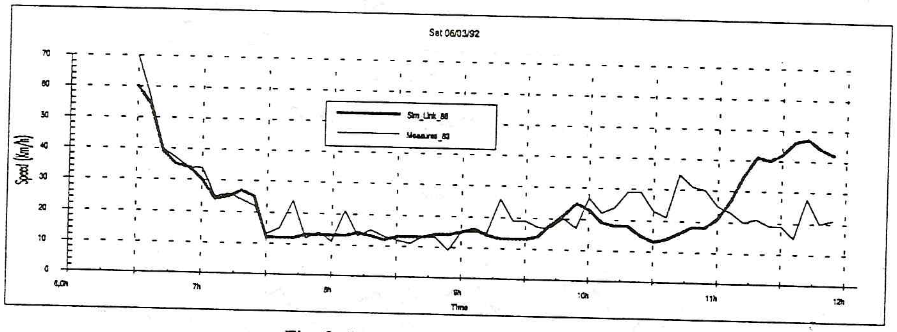


Fig. 8. Speeds of the motorway link 88

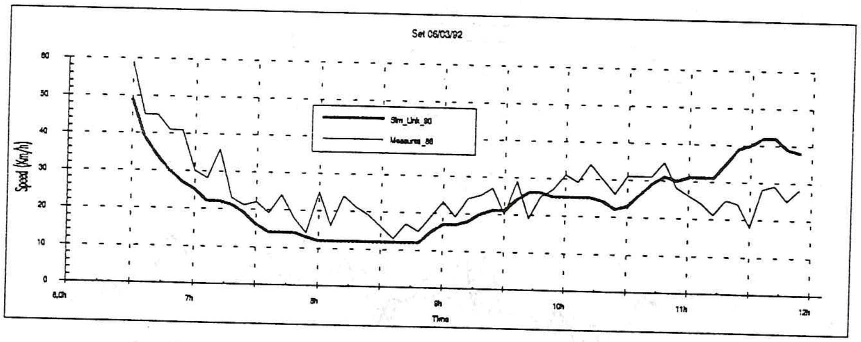


Fig. 9. Speeds of the motorway link 90

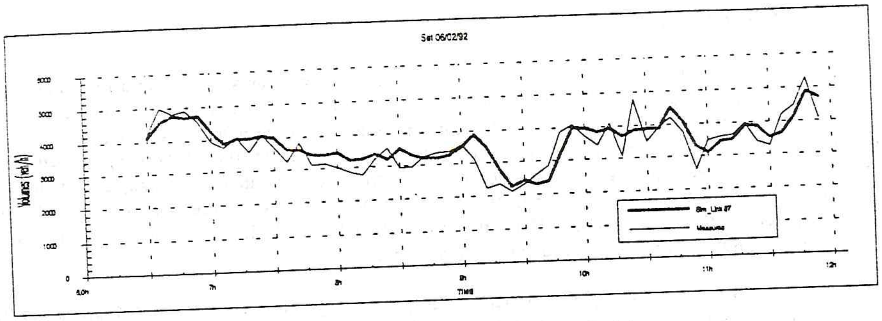


Fig. 10. Volumes of the motorway link 87

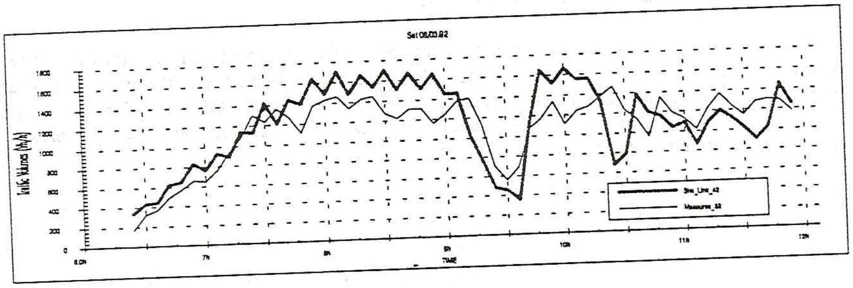


Fig. 11. Volumes of the urban link 42

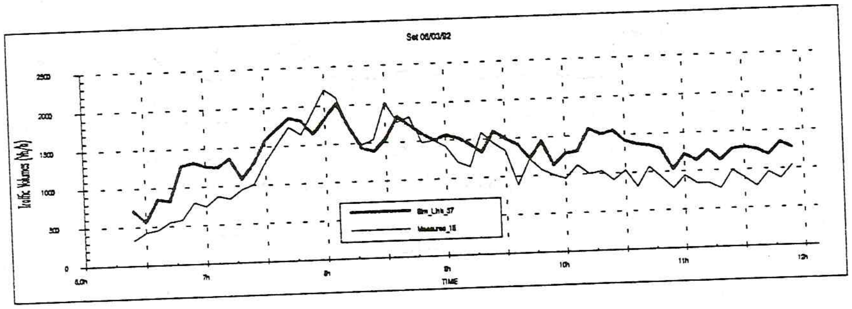


Fig. 12. Volumes of the urban link 57

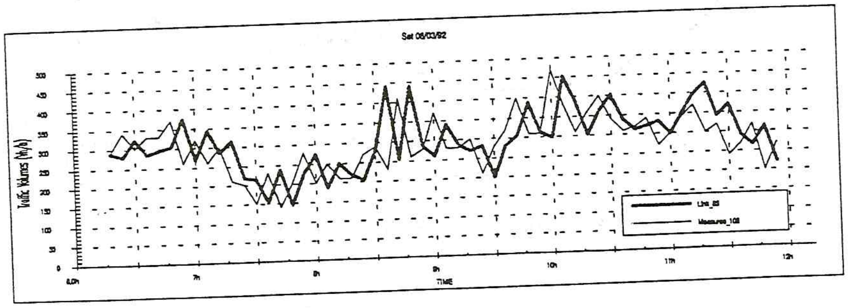


Fig. 13. Volumes of the urban link 83

V. CONCLUSIONS

METACOR has been succesfully applied to the Corridor Périphérique test site as indicated in the figures 10 to 13 which include only 6 stations among 30 available stations. Model parameters have been identified on one set of the real data (11/05/92) with some difficulties especially concerning the non-equipped urban origins which are considered as input variables of the model. The validation has been made respectively on the 03/06/92 and 02/06/92 real data sets with the same identified parameters used for 11/05/92.

The validation results concerning the output trajectories of the 3 traffic variables (volume, occupancy rates and speed) indicate that METACOR follows with sufficient accuracy the time evolution of the traffic conditions in the considered network. This means that METACOR is able to cope with different traffic conditions (fluid, dense and congested) at an acceptable level of accuracy. In more quantitative terms, calculation of the mean square deviation between the measured and the simulated variables was used as an indicator of accuracy. For example, on the urban part, the mean square deviation at the various measurement stations varies from 200 to 1000 veh/h whereas the mean values are varying from 1400 to 2800 veh/h.

As a final conclusion, METACOR is a generally applicable macroscopic simulation tool for corridor traffic. Compared to other available tools, its main advantage is due to the moderate computational effort. This advantage is the result of a systematic macroscopic approach with destination-specific subflows that avoids iterations for dynamic traffic assignment.

Acknowledgements

This work was conducted with support of the programme Advanced Transport Telematics of the European Commission, Project EUROCOR (V 2017). The authors acknowledge the support of Ville de Paris related to the data collection, data selection and data processing. The authors are thankful to C. Lambolet, G. Delthil, L.M Dupressoir, and the Police of Ville de Paris for valuable discussions during the calibration and validation procedure.

REFERENCES

[1] DRIVE I Project V 1035, CHRISTIANE, Deliverable N°6: "Network Modelling and Control", Dec 1989.

[2] DRIVE I Project V 1035; CHRISTIANE, Deliverable N°10: 'Software for traffic flow Modelling on Network Motorway", Nov. 1990.

[3] Messmer, A., Papageorgiou, M. : METANET : A macroscopic simulation program for motorway networks. Traffic Engineering & Control 31 (1990), No. 8/9, pp. 466 470.

[4] Papageorgiou, M.: Dynamic modeling, assignment, and route guidance in traffic networks. Transportation Research 24B (1990), No. 6, pp. 471 495.

[5] H. HAJ SALEM, " Gestion dynamique d'itinéraires urbains- reconstitution et prevision des temps de parcours", Paris XI university, PHD report, March 1984.

[6] Lebacque , J.P., "Simulation semi-macroscopique des réseaux urbains", Internal report, INRETS, Arcueil, 1983.

[7] PAPAGEORGIOU, M., BLOSSEVILLE, J.M., HAJ SALEM, H.: "Modelling and realtime control of traffic flow on the southern part of Boulevard Périphérique in Paris Part I: Modelling. Transportation Research 24A (1990), No. 5, pp. 345 359.

[8] Lebacque , J.P., "Semimacroscopic Simulation of urban Traffic"; Int 84 Minneapolis, AMSE 1984.

[9] DRIVE II Project V2017, EUROCOR, Deliverable N°4: "Integrated Model Development: Application to the Corridor Périphérique test site", Sept 93.

Computer Aided Design of Industrial Logistics Systems

Marc Goetschalckx, Ph.D.
George Nemhauser, Ph.D.
Michael H. Cole
Ru-Pei Wei
Koray Dogan
Xiaowei Zhang

Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2317

Version 1.1, May 24, 1994.

This research has been supported in part by the member companies of the Material Handling Research Center of the Georgia Institute of Technology.

Abstract

Recent geopolitical changes, the continuing development of truly global corporations, and the increasing importance of make-to-order manufacturing and inventory reductions in distribution systems have placed further emphasis on the growing role of logistics in the current business environment. One of the needs most often identified by companies is a tool to rapidly evaluate and design different logistics philosophies and configurations. This tool must be comprehensive, i.e. include all relevant costs and constraints, integrated, i.e. cover the logistics from supplier to final customer, systematic, i.e. provide high quality designs, and support easy manipulation and sensitivity analysis.

In this manuscript we report our continuing research in developing a comprehensive model for the design of integrated logistics systems and its associated solution techniques. In particular, we will focus on SMILE, a mixed integer programming algorithm to solve this model to optimality. Simultaneously, we also continue to develop the CIMPEL (Computer Integrated Modeling and Planning Environment for Logistics) tool, which allows easy user interaction, sensitivity analysis, and graphical displays. We will also comment on some recent case studies executed with this model, solution algorithm, and tool.

1. Introduction

Logistics is concerned with the movement, storage, and control of material from the suppliers, through the manufacturing and distribution facilities, all the way to the customer. As such it is one of the central functions of any production or manufacturing organization. Logistics encompasses many other disciplines such as facilities design, production and inventory control, and distribution and transportation. Because of the many interrelated subfunctions it is hard to measure the true cost of logistics and to design effective logistics systems.

In recent years several factors have contributed to the rapid changes in the area of logistics. Many corporations have developed a truly global view, where purchasing, production, and distribution are located in the most economical place anywhere in the world. Several major geopolitical changes occurred which opened new markets in Russia, China, and Eastern Europe. Increasing pressure to achieve the greater efficiency necessary for corporate survival and the associated business reengineering has led to a large number of reorganizations, acquisitions, and divestitures. Finally, lean and rapid response manufacturing generated more frequent and smaller production quantities with vastly reduced lead times.

The result of all these changes is that management has recognized distribution and logistics as a competitive tool to increase service and decrease cost. The above factors have also created the need for a continuing, integrated review and redesign of the logistics systems used by the corporation to adapt to the constantly changing business environment.

Traditionally, the design of strategic logistics and distribution systems has followed the four phases shown in the Figure 1. First, the customer service requirements are established to support the organization's business goals and strategy and marketing plans. Service levels may take the form of percentage delivered out of stock. Second, the materials deployment strategy is developed, which answers such questions as make-or-buy and in which manufacturing plant the goods will be produced. Third, a distribution network strategy is developed. Typical decisions are the number and location of distribution centers. Finally, a transportation strategy is established which allocates customers to distribution centers and selects transportation modes and carriers. It should be observed that we do not consider the operational decisions such as which truck and driver will serve a set of customers and in what order. Such problems belong to the class of vehicle routing problems and an recent overview is given in Golden and Assad (1989). Early papers on the network design problem were written by Geoffrion and Graves (1974) and Geoffrion et al (1974). An overview of such models is provided in Aikens (1985). Simultaneously, a large body of literature has appeared on inventory decisions. An overview is given in Silver and Peterson (1985).

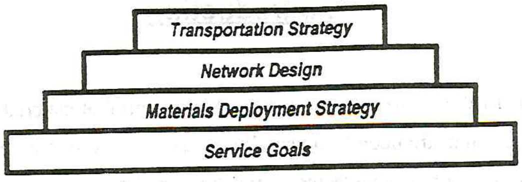


Figure 1. Traditional Logistics Design Phases

But, researchers realized that focussing exclusively on one of the above decision levels leads to suboptimization of the overall design problem. Several researchers integrated two levels of the above pyramid and extended the early models. Blumenfeld et all (1985), Bookbinder and Reece (1988), and Larson (1988) analyzed the tradeoffs between transportation and inventory costs.

The gains in productivity derived from integrated logistics systems can be substantial. Kearney (1984) reports that industrial corporations saved more than 12 % when they went to an integrated approach of design and operation of distribution systems. Perl and Sirisponsilp (1988) attempted for the first time to build a truly comprehensive model for the design of strategic distribution systems. They identified the interrelating factors in the areas of transportation, inventory, and location and proposed a model. No solution methods or computational results were reported.

The strategic goal of this research is thus to develop a basic understanding, design methodology, and decision making theory for integrated logistics systems. A schematic illustration of this process is given in the Figure 2. Here materials deployment, network design, and transportation strategy are all combined into a single integrated logistics model. Such models tend to be very large and very complex and their optimal solution is mathematically difficult. In previous research we developed a heuristic solution algorithm, called LAIR, see Song et all (1990). In this research we have developed an optimal solution based on mixed integer programming algorithms, denoted by SMILE.

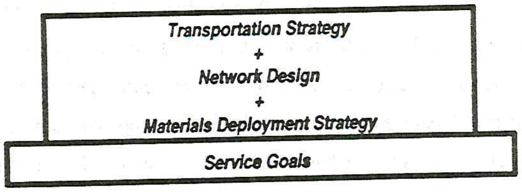


Figure 2. Modern Integrated Logistics Design

2. Integrated Logistics Model

Conceptually, the various logistics activities often can be viewed as an abstract network of nodes and arcs. The nodes represent facilities, points where product flows are created, processed, stored, and consumed (suppliers, plants, distribution centers, warehouses, customers). The arcs which are also called transportation channels, represent the flows or movements of products between the various facilities, or nodes. There may be several transportation channels between a pair of facilities to represent alternate form of transportation means, different shipping frequencies, and different cost structures.

The model as presented in this paper considers all material movement and storage from the moment the material leaves the manufacturing plant until it arrives at the customer. The model can handle multicommodity (i.e. more than one product group), multiecholon (i.e. zero, one, or more distribution centers between plant and customer), multichannel (i.e. alternative transportation channels between an origin and destination facility) problems. We have restricted ourselves to static or one period deterministic models (i.e. best, worst or average scenarios or future time period scenarios have to be created independently). Single sourcing of customers can be incorporated. The model incorporates production costs. The model has fixed costs associated with establishing or closing any facility. In addition, the model allows one step assembly operations in each of the distribution centers.

The remainder of this section lists basic assumptions, costs, and decisions associated with the various components of a strategic logistics system.

2.1. Model Components

2.1.1. Products

A set of products is given as input. All the products have the same basic unit load, e.g., pallet, truck load, or car load. Each product has value, weight, and volume per unit load. The value, weight, and volume per unit load of a product are constant throughout the logistics network. A product may be a combination of several different products; the model allows one step assembly operations in each of the warehouses. The part structure or bill of materials is limited to at most one level. That is, a product assembled from various component products cannot itself be used as a component product. The number of component products required to assemble a final product is prespecified. Total assembly cost is proportional to the number of final products assembled.

2.1.2. Plants

A set of potential production plants is given as input. Each plant has a location, a maximum production resource capacity, a fixed facility cost, and produces at a steady rate. A constant proportional cost is incurred for each unit load of product produced at a plant. The production cost per unit may differ among plants and products. Some plants may not produce some products. A constant proportional resource is consumed for each unit load of product produced at a plant. The resource requirement per unit may differ among plants and products. In addition, finished goods kept at a plant incur inventory holding costs according to a given holding cost rate. The production decisions are which plants to open and how much to produce for each product at each plant.

2.1.3. Warehousing

A set of potential warehouses is given. Any number of warehouses between plants and customers is possible, i.e., the system can be multiechelon. Associated with each potential warehouse are a location, maximum handling and storage capacities, and fixed and variable facility costs. All the different products are handled and stored in the same basic unit load; e.g., palletes or truck loads, and all the product flows, warehouse capacities, warehouse storage costs, and warehouse handling costs are expressed in terms of the basic unit load.

Since the problem is strategic and computing the cycle inventory in a multicommodity inventory system requires determining the phasing of inventory_replenishments, reordering cycle time, and lot sizes (Hall, 1988), the cycle inventory will be treated with approximations. It is assumed that each warehouse has many uncorrelated, unscheduled, and independent demands, so the product outflow of each warehouse is approaximately constant per fundamental time period and the product inflow is replenished simultaneously. The amount of safety stock is proportional to the throughput to model the inventory philosophy that a user determined number of periods of demands (safety stock factor) are held for safety stock. Each warehouse may perform assembly operations to combine different products into one another. There is no capacity limit on assembly operations. All assembly operations are performed instantaneously before products leave the warehouse. Warehousing decisions comprise which warehouses to open and the amount of product flow through each warehouse.

2.1.4. Transportation

Transportation can be divided into trunking and local delivery operations. Trunking comprises direct shipments from plants to warehouses, and from higher echelon (central) warehouses to lower echelon (local) warehouses. Local delivery comprises shipments from plants (direct shipment) or warehouses to customers. Local delivery options include direct shipment, peddling, and zoned delivery. The cost structure differs for each option. Since the problem is strategic, solving this strategic problem does not entail designing vehicle routes for local warehouses, which is a tactical or operational concern. Local delivery costs will be treated with approximations or as given. Each transportation channel has a shipping interval, deterministic travel time, travel distance, maximum allowable number of carriers, minimum number of required carriers, carrier weight capacity, carrier volume capacity, a cost per carrier, and a cost per unit load shipped. Pipeline inventories incur holding costs according to a given holding cost rate. Transportation decisions include allocation of customers to facilities, channel selection and the amount of each shipment.

2.1.5. Customers

A set of customers is given as input. Customers have deterministic demands for each of the products. Customer demands are steady over time, and do not depend on the level of service provided. Each customer is served every day. Some customers may require being served from a single facility, which is called the single sourcing requirement. Otherwise, customer orders may be split over different facilities and different transportation channels. The customer service requirements considered in current model are customer demand satisfaction, warehouse safety stock, single sourcing, and a maximum travel distance limitation between a customer and the warehouse that serves it.

The distribution centers have fixed costs, variable costs for handling different commodities and also variable storage costs for storing commodities. Transportation channels are capacitated both by weight and volume. Transportation costs are in terms of carrier containers such as trucks, railroad boxcars etc.., and/or proportional to the amount of material shipped, and/or proportional to the amount of material shipper per distance unit. The model also accounts for all inventory costs throughout the system. The amount of inventory in distribution centers is proportional to the throughput to model the inventory philosophy that a user determined number of periods of demands are held in inventory.

2.2. Verbal Formulation

Minimize total cost =

warehouse inventory cost	(Z1)
+ warehouse assembly cost	(Z2)
+ warehouse facility cost	(Z3)
+ plant production cost	(Z4)
+ plant inventory cost	(Z5)
+ plant facility cost	(Z6)
+ shipment cost	(Z7)
+ transportation inventory cost	(Z8)

Subject to:

flow conservation	(FC)
general flow conservation	(GF)
warehouse flow capacity	(TH, IF)

warehouse storage capacity	(ST, IS)
one warehouse type per site	(DT)
plant production capacity	(SP)
minimum/maximum number of carriers	(IC, XC)
carrier weight/volume capacity	(WE, VO)
ship to/from open warehouse only	(TD,FD)
ship from open plant only	(FP)
customer single sourcing	(SS1, 2, 3, 4)
max customer to warehouse distance	(MD)
customer demand satisfaction	(DM)
restrictions on variable values	

2.3. Notation

For consistency, all time-based parameters and variables are described in terms of days. Another fundamental time period can be substituted everywhere for the days.

2.3.1. Sets and Indexes

B : set of manufacturing facilities or plants, indexed by i

C : set of customers, indexed by k

CS : set of customers with single sourcing requirements ($CS \subseteq C$), indexed by k

D : set of warehouses or depots, indexed by j

L : set of warehouse types or sizes (e.g., small, medium, large, public), indexed by l

M : set of different transportation channels between a pair of facilities, indexed by m

P : set of products, indexed by p

PN : set of products not involved in assembly operations ($PN \subseteq P$), indexed by p

PC : set of all the component products of all the assembly operations ($PC \subseteq P$), indexed by p

PF : set of all final products of all the assembly operations ($PF \subseteq P$), indexed by f

PFp : set of all final products that the component product p may assemble ($PFp \subseteq PF$), indexed by f

2.3.2. Parameters

$PCap_i$: production capacity of plant i (resource units per day)

PFC_i : fixed cost of plant i ($ per day)

PPC_{ip} : production cost rate of plant i to produce one unit load of product p ($ per unit load)

PRU_{ip} : production resource units required for plant i to produce one unit load of product p (resource units per unit load)

WFC_{jl} : fixed cost of type l warehouse at site j ($ per day)

WHC_{jl} : handling cost of type l warehouse at site j ($ per unit load flow)

$WHCap_{jl}$: maximum handling capacity of type l warehouse at site j (unit loads per day)

WSC_{jl} : storage capacity cost of type l warehouse at site j ($ per unit load storage capacity per day)

$WSCap_{jl}$: maximum storage capacity of type l warehouse at site j (unit loads)

ssf_{jp} : safety stock factor of product p at warehouse j

AC_f : cost to assemble one unit load of final product f ($ per unit load)

AQt_{pf} : number of unit loads of component product p required to assemble one unit load of final product f

DM_{kp} : demand mean of customer k for product p (unit loads per day)

$MaxDistance_k$: maximum allowable distance from customer k to its server warehouse (miles)

$Distance_{jkm}$: travel distance on transportation channel jkm (miles)

T_{ijm} : transport time on transportation channel ijm (days)

R_{ijm} : time between shipments on transportation channel ijm, i.e., order interval (days)

TCC_{ijm} : shipment cost per carrier on transportation channel ijm ($ per carrier shipment)

TUC_{ijm} : shipment cost per unit load on transportation channel ijm ($ per unit load shipment)

$MinC_{ijm}$: minimum number of carriers required per order interval on transportation channel ijm (carriers per order interval)

$MaxC_{ijm}$: maximum allowable number of carriers per order interval on transportation channel ijm (carriers per order interval)

$CWCap_{ijm}$: carrier weight capacity of transportation channel ijm (weight units per carrier)

$CVCap_{ijm}$: carrier volume capacity of transportation channel ijm (volume units per carrier)

Val_p : value of a unit load of product p ($ per unit load)

Wt_p : weight of a unit load of product p (weight units per unit load)

Vol_p : volume of a unit load of product p (volume units per unit load)

r : inventory carrying cost rate ($ per $ per day)

2.3.3. Decision Variables

x_{ijmp} : average amount of product p shipped over transportation channel ijm (unit loads per day)

v_{jlp} : average amount of product p flow through warehouse of type l at site j (unit loads per day)

u_{jlp} : maximum amount of product p stored at warehouse of type l at site j (unit loads)

w_{ijm} : number of carriers used in one shipment from source facility i to destination facility j via transportation mode m

b_i : (0,1) 1 if plant is opened at site i, 0 otherwise

y_{jl} : (0,1) 1 if type l warehouse is opened at site j, 0 otherwise

z_{ijm} : (0,1) 1 if there is an open channel from source facility i to destination facility j via transportation mode m, 0 otherwise

q_{jk} : (0,1) 1 if customer k is served from facility j, 0 otherwise

2.4. Detailed Development

2.4.1. Warehousing

It is assumed that every R days, each warehouse places an order for a product. The sketch below shows an idealized single product inventory cycle.

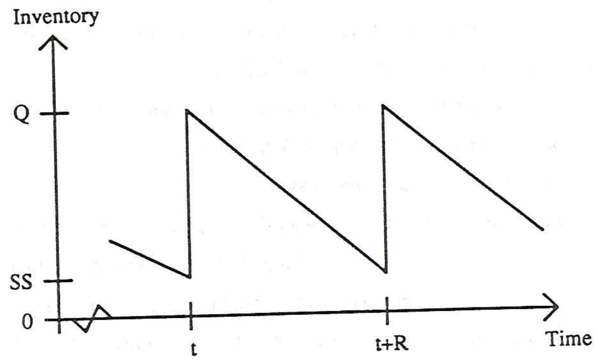


Figure 3. Idealized single product inventory cycle

Warehouse j placed an order and after the lead-time the order is received at time t. The inventory reach its highest level Q (Q = SS + demand rate × R) at time t. The cycle repeats after the next order is placed. The next order will be received at time t+R.

The warehouse inventory cost is simply the product of the inventory holding cost rate and the value of the average inventory. The mathematical form is developed below.

R = order interval

r = inventory holding cost rate

D_{jp} = demand rate for product *p* at warehouse *j*

$$D_{jp} = \sum_{i \in B \cup D} \sum_{m \in M} x_{ijmp} = \sum_{k \in C \cup D} \sum_{m \in M} x_{jkmp}$$

For convenience of modeling, it's assumed that the products are replenished simultaneously.

Q_{jp} = expected maximum inventory of product *p* at warehouse *j*

SS_{jp} = safety stock of product *p* at warehouse *j*

$$Q_{jp} = SS_{jp} + \sum_{i \in B \cup D} \sum_{m \in M} R_{ijm}\, x_{ijmp}$$

Safety stock is the product flow times a preset safety stock factor, e.g. three days of product flow.

$$SS_{jp} = \sum_{i \in B \cup D} \sum_{m \in M} ssf_{jp}\, x_{ijmp}$$

$$Q_{jp} = \sum_{i \in B \cup D} \sum_{m \in M} ssf_{jp}\, x_{ijmp} + \sum_{i \in B \cup D} \sum_{m \in M} R_{ijm}\, x_{ijmp}$$

AI_{jp} = average inventory of product p at warehouse j

$$AI_{jp} = \sum_{i \in B \cup D} \sum_{m \in M} ssf_{jp}\, x_{ijmp} + \left(\frac{1}{2}\right) \sum_{i \in B \cup D} \sum_{m \in M} R_{ijm}\, x_{ijmp}$$

Thus, the warehouse inventory cost for warehouse j is

$$\sum_{p \in P} r\, Val_p \left[\sum_{i \in B \cup D} \sum_{m \in M} ssf_{jp}\, x_{ijmp} + \left(\frac{1}{2}\right) \sum_{i \in B \cup D} \sum_{m \in M} R_{ijm}\, x_{ijmp} \right]$$

and the total system wide warehouse inventory cost is

$$\sum_{j \in D} \sum_{p \in P} r\, Val_p \left[\sum_{i \in B \cup D} \sum_{m \in M} ssf_{jp}\, x_{ijmp} + \left(\frac{1}{2}\right) \sum_{i \in B \cup D} \sum_{m \in M} R_{ijm}\, x_{ijmp} \right] \qquad (Z1)$$

This section develops the model of the assembly operations at warehouses. Moiseenko (1984) presents a generalized multicommodity network flow model which allows conversion of different commodities. Still, commodity conversion can not model assembly operations; the product assembly is a more general problem.

For those products that are not involved in any assembly operations, constraint (FC) enforces conservation of flow at each warehouse.

$$\sum_{i \in B \cup D} \sum_{m \in M} x_{ijmp} = \sum_{k \in C \cup D} \sum_{m \in M} x_{jkmp} \qquad j \in D, p \in PN \qquad (FC)$$

However, flow conservation does not hold true for assembly operations. Consider the following example where two units of product A and three units of product B are combined to form one unit of product C; two units of product B and one unit of product D are combined to form one unit of product E. The typical assembly equations for the operations are:

$$2\,A + 3\,B \rightarrow C$$

$$2\,B + 1\,D \rightarrow E$$

In this case, the flow conservation condition must be replaced by some more general constraints. Let x denote the input flows and y denote the output flows. The general flow conservation equations for the warehouse assembly operations are

$$x_A - y_A = 2\,(y_C - x_C)$$

$$x_B - y_B = 3\,(y_C - x_C) + 2\,(y_E - x_E)$$

$$x_D - y_D = (y_E - x_E)$$

The general flow conservation conditions for warehouse assembly operations are as follows:

$$\sum_{i\in B\cup D}\sum_{m\in M} x_{ijmp} - \sum_{k\in C\cup D}\sum_{m\in M} x_{jkmp} = \sum_{f\in PF_p} AQt_{pf}\left(\sum_{k\in C\cup D}\sum_{m\in M} x_{jkmf} - \sum_{i\in B\cup D}\sum_{m\in M} x_{ijmf}\right)$$

$$j\in D,\ p\in PC \qquad \text{(GF)}$$

The total warehouse assembly cost is

$$\sum_{j\in D}\sum_{f\in PF} AC_f\left(\sum_{k\in C\cup D}\sum_{m\in M} x_{jkmf} - \sum_{i\in B\cup D}\sum_{m\in M} x_{ijmf}\right) \qquad \text{(Z2)}$$

The facility cost of each warehouse includes a cost proportional to storage capacity, a cost proportional to the amount shipped through the warehouse, and a fixed cost. Each of these costs is defined in terms of a warehouse site and a warehouse type. In general, each potential warehouse site can be occupied by one of several types of warehouses. For example, small, medium, and large may be three possible types of warehouse. The notion of multiple warehouse types, each with a linear cost structure, allows a nonlinear (but piecewise linear) cost structure for each potential warehouse site.

The fixed cost of a warehouse accounts for overhead, capital costs, and such that are not considered to be proportional to storage capacity or throughput.

The various products are assumed to be handled and stored in the same unit loads, e.g., pallets, boxes, or trucks. All the product flows, warehouse capacities, warehouse storage costs, and warehouse handling costs are expressed in terms of the same unit loads.

The required storage capacity of a warehouse is determined by Q_{jp}, the previously derived maximum amount of inventory of product *p* stored at the warehouse.

The maximum amount of a product stored at a warehouse, u_{jlp}, is defined by the constraints (ST) and (IS). Constraint (ST) enforces the required warehouse storage capacity. Constraint (IS) states that the maximum amount stored at a warehouse is equal to the expected maximum inventory at a warehouse.

$$\sum_{p\in P} u_{jlp} \le WSCap_{jl}\, y_{jl} \qquad j\in D,\ l\in L \qquad \text{(ST)}$$

$$\sum_{l\in L} u_{jlp} = \sum_{i\in B\cup D}\ \sum_{m\in M} ssf_{jp}\, x_{ijmp} + \sum_{i\in B\cup D}\ \sum_{m\in M} R_{ijm}\, x_{ijmp} \quad j\in D,\ p\in P \qquad \text{(IS)}$$

The amount shipped through a warehouse, vjlp, is defined by the constraints (TH) and (IF). Constraint (TH) enforces warehouse handling capacity. Constraint (IF) states that the flow through a warehouse is equal to the flow into a warehouse.

$$\sum_{p\in P} v_{jlp} \le WHCap_{jl}\, y_{jl} \qquad j\in D,\ l\in L \qquad \text{(TH)}$$

$$\sum_{l\in L} v_{jlp} = \sum_{i\in B\cup D}\ \sum_{m\in M} x_{ijmp} \qquad j\in D,\ p\in P \qquad \text{(IF)}$$

Thus, the total warehousing facility cost is

$$\sum_{j\in D}\sum_{l\in L}\left(WFC_{jl}y_{jl} + WSC_{jl}\sum_{p\in P} u_{jlp} + WHC_{jl}\sum_{p\in P} v_{jlp}\right) \qquad \text{(Z3)}$$

Constraint (DT) requires that at most one type of warehouse be opened at a site.

$$\sum_{l\in L} y_{jl} \le 1 \qquad j\in D \qquad \text{(DT)}$$

2.4.2. Production

Plant costs include production costs, inventory costs, and fixed costs.

Plant production costs is proportional to each unit load of product produced at a plant, which is expressed in the following:

$$\sum_{i\in B}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\sum_{p\in P} PPC_{ip}x_{ijmp} \qquad \text{(Z4)}$$

Plant inventory costs:

$$\left(\frac{1}{2}\right)\sum_{i\in B}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\ \sum_{p\in P} r\ \mathrm{Val}_p\ R_{ijm}\ x_{ijmp} \qquad \text{(Z5)}$$

Plant fixed costs:

$$\sum_{i\in B} \mathrm{PFC}_i b_i \qquad \text{(Z6)}$$

Constraint (SP) prevents a plant from shipping product at a rate greater than its production capacity.

$$\sum_{j\in C\cup D}\ \sum_{m\in M}\ \sum_{p\in P} \mathrm{PRU}_{ip}\ x_{ijmp} \ \le\ \mathrm{PCap}_i \qquad i\in B \qquad \text{(SP)}$$

2.4.3. Transportation

Shipment costs comprise a variable cost proportional to the number of carriers used, the amount shipped, or both.

$$\sum_{i\in B\cup D}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\left(\frac{\mathrm{TCC}_{ijm}}{R_{ijm}}\right) w_{ijm} + \sum_{i\in B\cup D}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\ \sum_{p\in P} \mathrm{TUC}_{ijm} x_{ijmp} \qquad \text{(Z7)}$$

Transportation inventory costs are

$$\sum_{i\in B\cup D}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\ \sum_{p\in P} r\ \mathrm{Val}_p\ T_{ijm} x_{ijmp} \qquad \text{(Z8)}$$

Each transportation channel has a required minimum number of carriers used (maybe zero), which is the enforced by constraint (IC),

$$w_{ijm} \ \ge\ \mathrm{MinC}_{ijm} z_{ijm} \qquad i\in B\cup D,\ j\in C\cup D,\ m\in M \qquad \text{(IC)}$$

and a limited maximum allowable number of carriers used (maybe infinity), enforced by constraint (XC).

$$w_{ijm} \ \le\ \mathrm{MaxC}_{ijm} z_{ijm} \qquad i\in B\cup D,\ j\in C\cup D,\ m\in M \qquad \text{(XC)}$$

Each carrier has a weight capacity, enforced by constraint (WE),

$$\sum_{p\in P} R_{ijm} \mathrm{Wt}_p x_{ijmp} \ \le\ \mathrm{CWCap}_{ijm} w_{ijm} \qquad i\in B\cup D,\ j\in C\cup D,\ m\in M \qquad \text{(WE)}$$

and a volume capacity, enforced by constraint (VO).

$$\sum_{p\in P} R_{ijm} Vol_p x_{ijmp} \le CVCap_{ijm} w_{ijm} \qquad i \in B \cup D, j \in C \cup D, m \in M \qquad \text{(VO)}$$

Constraints (TD) and (FD) require that transportation channels to and from a potential depot site are usable only if a depot is actually opened at that site.

$$z_{ijm} \le \sum_{l\in L} y_{jl} \qquad i \in B \cup D, j \in D, m \in M \qquad \text{(TD)}$$

$$z_{jkm} \le \sum_{l\in L} y_{jl} \qquad j \in D, k \in C \cup D, m \in M \qquad \text{(FD)}$$

Constraint (FP) requires that transportation channels from a plant are usable only if a plant is open.

$$z_{ijm} \le b_i \qquad i \in B, j \in C \cup D, m \in M \qquad \text{(FP)}$$

Constraints (SS1) enforces that the customers requiring single sourcing be served by exactly one source.

$$\sum_{j\in B\cup D} q_{jk} = 1 \qquad k \in CS \qquad \text{(SS1)}$$

Constraints (SS2 and SS3) require that a customer can only be served from an open plant or an open warehouse.

$$q_{ik} \le b_i \qquad i \in B, k \in CS \qquad \text{(SS2)}$$

$$q_{jk} \le \sum_{l\in L} y_{jl} \qquad j \in D, k \in CS \qquad \text{(SS3)}$$

Constraint (SS4) requires that transportation channels from a source to a customer are usable only if the customer is served by that source.

$$z_{jkm} \le q_{jk} \qquad j \in B \cup D, k \in CS, m \in M \qquad \text{(SS4)}$$

Constraint (MD) requires that each customer be within a specified maximum travel distance from the warehouse or warehouse that serves it.

$$\sum_{j\in B\cup D} \sum_{m\in M} Distance_{jkm} z_{jkm} \le MaxDistance_k \qquad k \in C \qquad \text{(MD)}$$

2.4.4. Customers

Constraint (DM) ensures demand satisfaction.

$$\sum_{j\in B\cup D}\ \sum_{m\in M} \mathrm{x}_{jkmp} = \mathrm{DM}_{kp} \qquad k \in C, p \in P \qquad \text{(DM)}$$

2.4.5. Decision Variable Values

The values obtained by the decision variables are constrained as follows.

$$\mathrm{v}_{jlp} \geq 0 \qquad j \in D, l \in L, p \in P$$

$$\mathrm{u}_{jlp} \geq 0 \qquad j \in D, l \in L, p \in P$$

$$\mathrm{x}_{ijmp} \geq 0 \qquad i \in B \cup D, j \in C \cup D, m \in M, p \in P$$

$$\mathrm{w}_{ijm} \geq 0 \qquad i \in B \cup D, j \in C \cup D, m \in M$$

$$\mathrm{b}_i \in \{0,1\} \qquad i \in B$$

$$\mathrm{y}_{jl} \in \{0,1\} \qquad j \in D,\ l \in L$$

$$\mathrm{z}_{ijm} \in \{0,1\} \qquad i \in B \cup D, j \in C \cup D, m \in M$$

$$\mathrm{q}_{jk} \in \{0,1\} \qquad j \in D,\ k \in CS$$

2.5. Mathematical Formulation

Minimize

$$\sum_{j\in D}\sum_{p\in P} \mathrm{r\ Val}_p \left[\sum_{i\in B\cup D}\ \sum_{m\in M} \mathrm{ssf}_{jp}\ \mathrm{x}_{ijmp} + \left(\frac{1}{2}\right) \sum_{i\in B\cup D}\ \sum_{m\in M} \mathrm{R}_{ijm}\ \mathrm{x}_{ijmp} \right] \qquad \text{(Z1)}$$

$$\sum_{j\in D}\sum_{f\in PF} \mathrm{AC}_f \left(\sum_{k\in C\cup D}\ \sum_{m\in M} \mathrm{x}_{jkmf} - \sum_{i\in B\cup D}\ \sum_{m\in M} \mathrm{x}_{ijmf} \right) \qquad \text{(Z2)}$$

$$\sum_{j\in D}\sum_{l\in L} \left(\mathrm{WFC}_{jl}\mathrm{y}_{jl} + \mathrm{WSC}_{jl} \sum_{p\in P} \mathrm{u}_{jlp} + \mathrm{WHC}_{jl} \sum_{p\in P} \mathrm{v}_{jlp} \right) \qquad \text{(Z3)}$$

$$\sum_{i\in B}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\sum_{p\in P} \mathrm{PPC}_{ip}\mathrm{x}_{ijmp} \qquad \text{(Z4)}$$

$$\left(\frac{1}{2}\right)\sum_{i\in B}\ \sum_{j\in C\cup D}\ \sum_{m\in M}\ \sum_{p\in P} \mathrm{r\ Val}_p\ \mathrm{R}_{ijm}\ \mathrm{x}_{ijmp} \qquad \text{(Z5)}$$

$$\sum_{i\in B} \mathrm{PFC}_i \mathrm{b}_i \qquad \text{(Z6)}$$

$$\sum_{i \in B \cup D} \sum_{j \in C \cup D} \sum_{m \in M} \left(\frac{\mathrm{TCC}_{ijm}}{\mathrm{R}_{ijm}} \right) \mathrm{w}_{ijm} + \sum_{i \in B \cup D} \sum_{j \in C \cup D} \sum_{m \in M} \sum_{p \in P} \mathrm{TUC}_{ijm} \mathrm{x}_{ijmp} \qquad \text{(Z7)}$$

$$\sum_{i \in B \cup D} \sum_{j \in C \cup D} \sum_{m \in M} \sum_{p \in P} \mathrm{r}\ \mathrm{Val}_p\ \mathrm{T}_{ijm} \mathrm{x}_{ijmp} \qquad \text{(Z8)}$$

Subject to:

$$\sum_{i \in B \cup D} \sum_{m \in M} \mathrm{x}_{ijmp} = \sum_{k \in C \cup D} \sum_{m \in M} \mathrm{x}_{jkmp} \qquad j \in D, p \in PN \qquad \text{(FC)}$$

$$\sum_{i \in B \cup D} \sum_{m \in M} \mathrm{x}_{ijmp} - \sum_{k \in C \cup D} \sum_{m \in M} \mathrm{x}_{jkmp} = \sum_{f \in PF_p} \mathrm{AQt}_{pf} \left(\sum_{k \in C \cup D} \sum_{m \in M} \mathrm{x}_{jkmf} - \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{x}_{ijmf} \right)$$

$$j \in D, p \in PC \qquad \text{(GF)}$$

$$\sum_{p \in P} \mathrm{v}_{jlp} \leq \mathrm{WHCap}_{jl}\ \mathrm{y}_{jl} \qquad j \in D, l \in L \qquad \text{(TH)}$$

$$\sum_{l \in L} \mathrm{v}_{jlp} = \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{x}_{ijmp} \qquad j \in D, p \in P \qquad \text{(IF)}$$

$$\sum_{p \in P} \mathrm{u}_{jlp} \leq \mathrm{WSCap}_{jl}\ \mathrm{y}_{jl} \qquad j \in D, l \in L \qquad \text{(ST)}$$

$$\sum_{l \in L} \mathrm{u}_{jlp} = \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{ssf}_{jp}\ \mathrm{x}_{ijmp} + \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{R}_{ijm}\ \mathrm{x}_{ijmp} \quad j \in D, p \in P \qquad \text{(IS)}$$

$$\sum_{l \in L} \mathrm{u}_{jlp} = \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{ssf}_{jp}\ \mathrm{x}_{ijmp} + \sum_{i \in B \cup D} \sum_{m \in M} \mathrm{R}_{ijm}\ \mathrm{x}_{ijmp} \quad j \in D, p \in P \qquad \text{(IS)}$$

$$\sum_{l \in L} \mathrm{y}_{jl} \leq 1 \qquad j \in D \qquad \text{(DT)}$$

$$\sum_{j \in C \cup D} \sum_{m \in M} \sum_{p \in P} \mathrm{PRU}_{ip}\ \mathrm{x}_{ijmp} \leq \mathrm{PCap}_i \qquad i \in B \qquad \text{(SP)}$$

$$\mathrm{w}_{ijm} \geq \mathrm{MinC}_{ijm} \mathrm{z}_{ijm} \qquad i \in B \cup D, j \in C \cup D, m \in M \qquad \text{(IC)}$$

$$\mathrm{w}_{ijm} \leq \mathrm{MaxC}_{ijm} \mathrm{z}_{ijm} \qquad i \in B \cup D, j \in C \cup D, m \in M \qquad \text{(XC)}$$

$$\sum_{p \in P} \mathrm{R}_{ijm} \mathrm{Wt}_p \mathrm{x}_{ijmp} \leq \mathrm{CWCap}_{ijm} \mathrm{w}_{ijm} \qquad i \in B \cup D, j \in C \cup D, m \in M \qquad \text{(WE)}$$

$$\sum_{p \in P} \mathrm{R}_{ijm} \mathrm{Vol}_p \mathrm{x}_{ijmp} \leq \mathrm{CVCap}_{ijm} \mathrm{w}_{ijm} \qquad i \in B \cup D, j \in C \cup D, m \in M \qquad \text{(VO)}$$

$$z_{ijm} \leq \sum_{l \in L} y_{jl} \qquad i \in B \cup D, j \in D, m \in M \qquad \text{(TD)}$$

$$z_{jkm} \leq \sum_{l \in L} y_{jl} \qquad j \in D, k \in C \cup D, m \in M \qquad \text{(FD)}$$

$$z_{ijm} \leq b_i \qquad i \in B, j \in C \cup D, m \in M \qquad \text{(FP)}$$

$$\sum_{j \in B \cup D} q_{jk} = 1 \qquad k \in CS \qquad \text{(SS1)}$$

$$q_{ik} \leq b_i \qquad i \in B, k \in CS \qquad \text{(SS2)}$$

$$q_{jk} \leq \sum_{l \in L} y_{jl} \qquad j \in D, k \in CS \qquad \text{(SS3)}$$

$$z_{jkm} \leq q_{jk} \qquad j \in B \cup D, k \in CS, m \in M \qquad \text{(SS4)}$$

$$\sum_{j \in B \cup D} \sum_{m \in M} \text{Distance}_{jkm} z_{jkm} \leq \text{MaxDistance}_k \qquad k \in C \qquad \text{(MD)}$$

$$\sum_{j \in B \cup D} \sum_{m \in M} x_{jkmp} = \text{DM}_{kp} \qquad k \in C, p \in P \qquad \text{(DM)}$$

$$v_{jlp} \geq 0 \qquad j \in D, l \in L, p \in P$$

$$u_{jlp} \geq 0 \qquad j \in D, l \in L, p \in P$$

$$x_{ijmp} \geq 0 \qquad i \in B \cup D, j \in C \cup D, m \in M, p \in P$$

$$w_{ijm} \geq 0 \qquad i \in B \cup D, j \in C \cup D, m \in M$$

$$b_i \in \{0,1\} \qquad i \in B$$

$$y_{jl} \in \{0,1\} \qquad j \in D, l \in L$$

$$z_{ijm} \in \{0,1\} \qquad i \in B \cup D, j \in C \cup D, m \in M$$

$$q_{jk} \in \{0,1\} \qquad j \in D, k \in CS$$

3. Optimal Solution Algorithm: SMILE

The above model belongs to the class of large scale mixed integer programming formulations. At the current we solve this model by generating the corresponding MPS file from the project data base and submitting this MPS file to MINTO (Savelsbergh & Nemhauser). MINTO is a mixed integer programming solver developed at the Georgia Institute of Technology which uses in turn CPLEX or OSL as linear programming solvers. The solution from MINTO is then inserted in project data base and then displayed by CIMPEL.

3.1. General Framework

The solution method presented here works in two phases. The first phase includes preprocessing and reformulation. The second phase includes three major components: branching rules, primal heuristics, and valid inequalities. This second branch and cut phase will be done by customizing MINTO application functions.

3.1.1. Preprocessing

Preprocessing includes simplifying rows and columns by identifying redundant rows, fixing variables, increasing lower and decreasing upper bounds on variables. Many detailed preprocessing techniques described in Savelsbergh (1992) are already implemented inside MINTO. The preprocessing approach presented here focuses on simplifying rows and columns, increasing lower and decreasing upper bounds on variables from the logical implications of the problem data. Following is a list of cases where the formulation may be simplified.

- The total product flow through an arc or a node can not exceed total customer demands in the system. If the flow capacity of a warehouse type exceeds the total customer demands, the flow capacity of that warehouse type can be replaced by the total customer demands. If the production capacity of a plant exceeds the largest possible production resource units used, the plant production capacity can be replaced by the largest possible production resource units used.

- Constraint (MD) requires that each customer be within a specified maximum travel distance from the warehouse that serves it. For those warehouses located farther than the specified travel distance from a customer, the transportation channels and the product flows between those warehouses and that customer can be eliminated from the formulation without affecting the solution.

3.1.2. Reformulation

A real world problem may not require all the features of the model. Many cost components and constraints may be optional depending on the problem instance. For example, if a customer does not require a single source, the single sourcing constraints (SS1 and SS2) for that customer will not appear in the formulation. There are several cases for which part of the formulation can be changed to achieve better solution performances.

- For an existing warehouse j, constraint (DT') requires that exactly one type of warehouse be opened.

- If a transportation channel ijm doesn't have minimum number of carrier restriction and the destination customer of that channel does not require single sourcing, the (XC), (IC), (TD), and (FD) constraints can be reformulated to eliminate the channel binary variable zijm. The new formulations are then

$$w_{ijm} \leq \text{MaxC}_{ijm} \qquad i \in B,\ j \in C,\ m \in M \qquad \text{(XC')}$$

$$w_{ijm} \leq \text{MaxC}_{ijm} \sum_{l \in L} y_{jl} \qquad i \in B \cup D,\ j \in D,\ m \in M \qquad \text{(TD')}$$

$$w_{jkm} \leq \text{MaxC}_{jkm} \sum_{l \in L} y_{jl} \qquad j \in D,\ k \in C \cup D,\ m \in M \qquad \text{(FD')}$$

$$w_{ijm} \leq \text{MaxC}_{ijm} b_i \qquad i \in B,\ j \in C \cup D,\ m \in M \qquad \text{(FP)}$$

- If the value of a particular commodity equals zero then all terms related to the inventory holding process for that commodity can be eliminated from the model.

3.1.3. Branching Rules

The proposed branching rules are listed as follows:

- First, branch on the plant binary variables bi. Within different plant binary variables, branch on the one with value closest to 0.5 first.

- Second, branch on the warehouse type binary variables yjl. Since for each warehouse site the warehouse type variables are special order sets, set all the warehouse variables with flow capacity larger than the warehouse internal flow to zero on a branch and set them to one on the other branch, i.e., one branch will be to build a warehouse of size larger than the current product flow, while the other branch will be to build a warehouse of size smaller than the current product flow. Within different warehouse sites, branch on the warehouse with the largest product flow.

- Third, branch on the single sourcing variables qjl. Since for each customer the single sourcing variables are also special order sets, follow the procedures similar to the warehouse type, but branch by transportation costs. Within different customers, branch on the customer with the largest demand.

- Forth, branch on the channel binary variables zjijm. within different channel binary variables, branch on the one with value closest to 0.5.

3.1.4. Primal Heuristics

The proposed primal heuristics for finding an upper bound of the optimal solutions is a linear programming based heuristics. Solve the LP relaxation problem; follow the order of plant, warehouse, customer, and channels in the branching rule section; fix one binary variable with value closest to one, except for the warehouse type variables fix the variable with capacity closest to the internal flow; then resolve the problem with LP. Iterate this procedure until there are no more binary variables with fractional values.

3.1.5. Valid Inequalities

Following is a set of valid inequalities that can be added to the constraints to tighten the formulation:

$$\sum_{m\in M} x_{jkmp} \le \sum_{l\in L} \mathrm{DM}_{kp}\, y_{jl} \qquad j\in D,\ k\in C,\ p\in P \qquad \text{(VI1)}$$

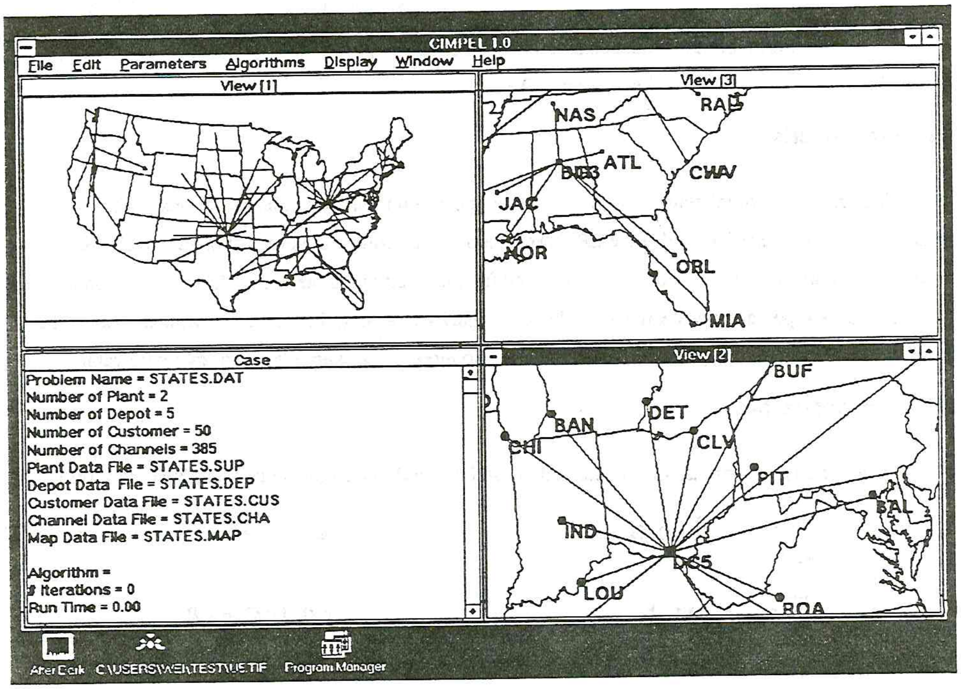
$$\sum_{m\in M} x_{ijmp} \le \mathrm{DM}_{jp}\, b_i \qquad i\in B,\ j\in C,\ p\in P \qquad \text{(VI1)}$$

$$\sum_{m\in M} x_{jkmp} = \mathrm{DM}_{kp}\, q_{jk} \qquad j\in B\cup D,\ k\in CS,\ p\in P \qquad \text{(VI1)}$$

Several case studies have been completed or are in progress using the SMILE algorithm to design a variety of logistics systems. Preliminary computational results are given in the section on case studies.

4. Logistics Design Environment: CIMPEL

CIMPEL was developed to run under the Microsoft Windows operating environment, which provides the graphical user interface. A typical display screen for the design of a distribution system of a company in the continental United States is shown in the following Figure. Multiple windows allow the user to focus on different aspects of the same problem, while still maintaining the overall view and numerical statistics.


CIMPEL 1.0
File Edit Parameters Algorithms Display Window Help
View [1]
View [3]
NAS
RAL
ATL
DEB
CWV
JAC
NOR
ORL
MIA
Case
View [2]
BUF
DET
BAN
CHI
CLV
PIT
BAL
IND
DC5
LOU
ROA
AfterDark C:\USERS\WEI\TEST\UE.TIF Program Manager

```
Problem Name = STATES.DAT
Number of Plant = 2
Number of Depot = 5
Number of Customer = 50
Number of Channels = 385
Plant Data File = STATES.SUP
Depot Data  File = STATES.DEP
Customer Data File = STATES.CUS
Channel Data File = STATES.CHA
Map Data File = STATES.MAP

Algorithm =
# Iterations = 0
Run Time = 0.00
```

Figure 4. CIMPEL Screen Illustration

The CIMPEL environment encompasses an algorithms toolkit which includes heursitics for rapid prototyping, optimal algorithms such as SMILE for detailed design, evaluation algorithms for sensitivity analysis, and simulation for verification. All algorithms use the same underlying data base, which corresponds to the integrated logistics model, so that their results are consistent and comparable.

5. Case Studies

At the current time several case studies are being conducted with the member companies of the Material Handling Research Center, but two studies have already been concluded. The first study involves the design of the distribution network for a national hospital supply company. The second case study involves the design of the system to recycle residential plastics in the state of New Jersey. The best case recycling system was 25 % less expensive than the current recycling system. Observe that the recycling study used the same logistics model as the distribution model without any modifications, except setting the inventory values to zero. Individual runs took less than 180 minutes on an IBM RISC 6000 workstation. We are currently in the process of reducing the required computation times, while at the same time retain the general model structure. The generated problem sizes, solution algorithm properties, and computation times are given in the next two tables

Table 1. Model Sizes and Computation Times for the Case Studies

Case	# Products	# Depots	# Cust.	# Plants	# Channels
New Jersey	1	64	21	1	1408
Ohio	4	3	8	5	45
Hospital Supplies	46	7	49	46	1029
Electronics	30	25	55	30	2125
Case	# Const.	# Vars.	# Binary	# Integer	CPU Time
New Jersey	2902	1472	64	0	38 sec
Ohio	209	270	9	45	30 min
Hospital Supplies	27441	15128	1122	0	179 min
Electronics	2919	68400	150	0	99 min

Table 2. MIP Solution Properties and Computation Times (Electronics Case Study)

Software	CPU	Nodes	Depth of	LPs	Cuts
	seconds	Evaluated	B&B Tree	Solved	Generated
Minto (i)	27428	309	114	334	412
Minto (i,f at root)	21482	229	111	257	379(i),1755 (f)
Minto (B&B only)	7103	399	114	399	
Minto (g)	5913	259	44	259	
CPLEX (def. MIP)	9630	663		663	
i: Implication Inequalities					
f: Generalize Flow Covers					
g: Primal Heuristics fixing vars.					

6. Conclusions

This research has shown that the development and solution of comprehensive models for the design of integrated logistics systems is possible. Deterministic models can be solved with current state of the art mixed integer programming techniques and solvers. The solution times on powerful workstations are significant but not unacceptable. An interactive graphical front end is required to communicate the results of this model and to allow an easy sensitivity analysis. Similarly, an consistent underlying data base is required to allow for the execution of different tools and algorithms. The availability of relatively inexpensive but powerful computer workstations graphical user interfaces encourages further developments in these three directions.

References

Aikens, C.H., "Facility location models for distribution planning," European Journal of Operations Research, 22, 263-279 (1985).

Ballou, R.H. Business Logistics Management, Prentice-Hall, Inc., Englewood Cliffs (1985).

Bhaskaran, S., and Turnquist, M.A., "Multiobjective transportation considerations in multiple facility location," Transportation Research- A, 24A, 2, 139-148 (1990).

Bilde, O., and Krarup, J., "Sharp lower bounds for the simple location problem," Annals of Discrete Mathematics, 1, 79-97 (1977).

Copacino, W., and Rosenfield, D.B., "Analytic tools for strategic planning," International Journal of Physical Distribution and Materials Management, 15, 3, 47-61 (1985).

Council of Logistics Management, "What it's all about," (1990).

Daganzo, C.F., Logistics Systems Analysis, Springer-Verlag, Berlin (1991).

Delaney, R.V., "Trends in logistics and U.S. world competitiveness," Transportation Quarterly, 45, 1, 19-41 (January 1991).

Erlenkotter, D., "A dual-based procedure for uncapacitated facility location," Operations Research, 26, 992-1009 (1978).

Gao, L., and Robinson Jr., E.P., "A dual based optimization procedure for the two-echelon uncapacitated facility location problem," Naval Research Logistics, 39, 191-212 (1992).

Geoffrion, A.M., and Graves, G.W., "Multicommodity distribution system design by Benders decomposition," Management Science 20, 5, 822-844 (1971).

Hall, N.G., "A multi-item EOQ model with inventory cycle balancing," Naval Research Logistics 35, 319-325 (1988).

Ho, P., "Warehouse location with service sensitive demand," Ph.D. Dissertation (J. Perl, chairman), University of Maryland (1989).

House, R.G., and Karrenbauer, J.J., "Logistics system modeling," International Journal of Physical Distribution and Materials Management, 12, 3, 119-129 (1982).

Kearney, A.T., Measuring and Improving Productivity in Physical Distribution, National Council of Physical Distribution Management, Oak Brook, Illinois (1984).

Klincewicz, J.G., and Luss, H., "A dual-based algorithm for multiproduct uncapacitated facility location," Transportation Science, 21, 3, 198-206 (1987).

Kuehn, A.A., and Hamburger, M.J., "A heuristic program for locating warehouses", Management Science, 9, 643-666 (1963)

Magee, J.F., Copacino, W.C., and Rosenfield, D.B., Modern Logistics Management, John Wiley & Sons, new York (1985).

Moiseenko, G.E., "A multicommodity network flow problem with commodity conversion and dependent flows," Translated from Avtomatika i Telemekhanika, 11, 93-100 (1984).

Novich, N.S., "Leading edge distribution strategies," The Journal of Business Strategy, 48-53 (November/December 1990).

Perl, J., and Sirisoponsilp, S., "Distribution networks: facility location, transportation, and inventory," International Journal of Physical Distribution and Materials Management, 18, 6, 18-26 (1988).

Savelsbergh, M.W.P., "Preprocessing and probing for mixed-integer programming problem," COSOR-92-06, Eindhoven University of Technology (1992).

Simchi-Levi, D., "Hierarchical planning for probabilistic distribution systems in Euclidean spaces," Management Science, 38, 2, 198-211 (1992).

Silver, E.A., and Peterson, R., Decision Systems for Inventory Management and Production Planning, Second Edition, John Wiley & Sons, New York (1985).

Sirisoponsilp, S., "Warehouse location under multiple transportation options," Ph.D. dissertation (J. Perl, chairman), University of Maryland (1989).

Tcha, D., and Lee, B. "A branch-and-bound algorithm for the multilevel uncapacitated facility location problem," European Journal of Operational Research, 18, 35-43 (1984).

Van Roy, T.J., and Erlenkotter, D., "A dual-based procedure for dynamic facility location," Management Science, 28, 1091-1105 (1982).

DECOMPOSITION ALGORITHMS FOR SOLVING OF SOME PRODUCTION-TRANSPORTATION PROBLEMS *

Alexander A. Kolokolov
Institute of Information Technologies and Applied
Mathematics, Russian Academy of Sciences,
28, Andrianov St., 644077, Omsk, Russia
e-mail: iitpm@intekh.omsk.su

1 Introduction

In [2]–[6] we develop an approach to investigation and solving of integer programming problems (IPP), which is based on the application of regular partitions of the space $\mathbf{R}^n$. With the help of L-partition and other ones we have introduced new classes of cuts, obtained bounds of iteration number for cutting plane algorithms (in particular, for first Gomory algorithm), for some branch-and-bound algorithms, suggested L-class enumeration and hybrid algorithms for IPP, got other theoretical and experimental results.

In this paper we develop decomposition L-class enumeration algorithms for some mixed integer production-transportation problems. An algorithm iteration includes solving of production and transportation subproblems. The Benders cuts are used for modification of the current integer production subproblem, which is solved by L-class enumeration algorithms and regular cuts. This method is applicable to discrete multicommodity production- transportation problems and other mixed integer problems. Earlier decomposition method for solving of such problems with boolean implicit enumeration was described in [1].

Consider the following production - transportation problem. There are n plants and m consumers of a commodity. Each plant has some variants of production and may transportate the commodity to any consumer. The variants of production are alternative. The values of consumption are known. It is necessary to find a plan of production and transportation of the commodity with minimal common cost of this operations.

Introduce the following notations:

i - index for the plant, $i = 1, \dots, n$,
j - index for the consumer, $j = 1, \dots, m$,
w_i - number of production variants for plant i,
r - index for plant variant, $r = 1, \dots, w_i$,
c_i^r - cost of variant r for plant i,
s_{ij} - unit transportation cost of the commodity from plant i to consumer j,
a_i^r - value of the commodity for variant r of plant i,
b_j - value of consumption for consumer j.

*This work was supported by the Russian Fund of Fundamental Research, project code 93-01-00557.

Let x_{ij} be the value of the commodity being transported from plant i to consumer j, and

$$z_i^r = \begin{cases} 1 & \text{, if we choose variant r for plant i,} \\ 0 & \text{, otherwise.} \end{cases}$$

Denote $x = (x_{ij}); z = (z_i^r); i = 1, \ldots, n; j = 1, \ldots, m; r = 1, \ldots, w_i$.
The mixed integer programming model is as follows:

$$f(z, x) = \sum_{i=1}^{n} \sum_{r=1}^{w_i} c_i^r z_i^r + \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} x_{ij} \rightarrow \min \tag{1}$$

s.t.

$$\sum_{r=1}^{w_i} a_i^r z_i^r - \sum_{j=1}^{m} x_{ij} \geq 0, \; i = 1, \ldots, n, \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} \geq b_j, \; j = 1, \ldots, m, \tag{3}$$

$$\sum_{r=1}^{w_i} z_i^r \leq 1, \; i = 1, \ldots, n, \tag{4}$$

$$z_i^r \in \{0, 1\}, \; r = 1, \ldots, w_i; \; i = 1, \ldots, n, \tag{5}$$

$$x_{ij} \geq 0, \; i = 1, \ldots, n; j = 1, \ldots, m. \tag{6}$$

Inequalities (2)-(3) are balance conditions for the plants and the consumers respectively, (4) are conditions of alternatives.

2 L-partition approach

Give some definitions. First introduce L-partition of space $\mathbf{R^n}$. Let $\succ$, $\succeq$ be symbols of the lexicographical order. Points x,$y \in \mathbf{R^n}$ $(x \succ y)$ are L-equivalent, if there is no $z \in \mathbf{Z^n}$ such that $x \succeq z \succeq y$. This equivalence generates a partition of any set $X \subseteq \mathbf{R^n}$. Elements of the partition are called L-classes. L-partition has some useful properties.

1) Every point $z \in \mathbf{Z^n}$ forms individual L - class. The other classes consist of noninteger points only and they are called fractional.

2) If X is bounded set, then X/L is finite.

3) L - partition is coordinated with lexicographic order, i.e. for each X all elements of X/L can be put in linear ordering: $V, V' \in X/L, V \succ V' \Leftrightarrow x \succ x'$ for all $x \in V, x' \in V'$.

If X is bounded, then

$$X/L = \{V_1, \ldots, V_q\}, V_i \succ V_{i+1}, i = 1, \ldots, q-1.$$

The number

$$r(V) = \begin{cases} min\{i : x_i \neq \lfloor x \rfloor, i = 1, \ldots, n\}, & \text{if V is fractional } L\text{-class;} \\ n + 1, & \text{otherwise,} \end{cases}$$

is called rank of L-class.

The L - class enumeration algorithm is based on the idea of the search the next L - partition element in the order of lexicographic increase.

Let $B^n = \{x : 0 \leq x_i \leq 1, \ i = 1, \ldots, n\}$. Consider this method for a polyhedron $M \subset B^n$. The boolean programming problem (BPP) is: to find

$$z_* = lexmin(M \cap \mathbf{Z^n}).$$

Corresponding linear programming problem (LPP) consists of the search of lexicographically minimal element of set M.

The set

$$M_* = \{x \in \ M : x \succ z \ \forall z \in \ M \cap \mathbf{Z^n}\}$$

is called a fractional covering of above BPP. If $M \cap \mathbf{Z^n} = \emptyset$, then $M_* = M$. The factor-set M_*/L is called L-covering. Let $V_{x^*}(\ M_*)$ be an element of M_*/L containing x^*. A linear inequality $(g, x) \leq g_0$ is called L-regular cut if

a) $(g, x') > g_0$ for any $x' \in V_{x^*}(\ M_*)$,

b) $(g, z) \leq g_0$ for any $z \in \ M \cap \mathbf{Z^n}$.

There are some methods for construction of such cuts. In particular, some Gomory cuts are L-regular [3]. The depth of the cut equals to number of L-classes excluded by it from L-covering of the IPP.

Let $V \in M/L$ and some representative $\bar{x} \in V$ is known. First, we seek neighboring fractional element V' to V so that $x_i = \bar{x}_i, \ i = 1, \ldots, r-1; x_r = \lceil \bar{x_r} \rceil$, where r is rank of class V, and x is certain point from V'. If V' will be obtained, then continue this process for V' instead of V.

Otherwise, we seek V' such that $x_i = \bar{x}_i \ , i = 1, \ldots, r'-1, \ , x_{r'} > \bar{x}_{r'}, r' = r-1$, where r' - is rank of V', $x \in V'$. If V' can't be obtained, we reduse (if it is possible) r' by 1 and continue search. If V' will be obtained, we rise over the process beginning and V' became initial L - class.

If there is no neighboring fractional L - class, we receive optimum of the BPP or decide, that problem has no solution. The process is finite because M is bounded.

Describe L - class enumeration algorithm . For simplicity an iteration number will be pass.

Step 0. Solve the initial LPP. If it has not feasible solution x or it's solution belong to $\mathbf{Z^n}$, process is finished. Otherwise go to step 1.

Step 1. Denote by $\bar{x}$ optimal solution of LPP from previous step . Find

$$p = \min\{i : \bar{x} \neq \lfloor x_i \rfloor, \ i = 1, \ldots, m\}.$$

Form LPP by the addition the following constraints to the initial problem

$$x_1 = \bar{x}_1, \ldots, x_{p-1} \ = \ \bar{x}_{p-1}, \, x_p \leq \lceil \bar{x}_p \rceil.$$

Its goal function is $(-x_p) \rightarrow$ lexmin. Find solution x' of this problem. It is possible:

1) $x' \in \mathbf{Z^n}$, then process is finished;

2) $x' \notin \mathbf{Z^n}$, then the variants may be:

a) $x'_p < \lceil \bar{x}_p \rceil$; if $p = 1$, then process is finished, otherwise go to step 2;

b) $x_p = \lceil \bar{x}_p \rceil$, then go to step 1.

Step 2. Find a maximal number $\psi \leq p-1$, such as $\bar{x}_\psi < 1$. Form LPP by the addition the following constraints to the initial problem

$$x_1 = \bar{x}_1, \ldots, x_{\psi-1} = \bar{x}_{\psi-1}, x_\psi \leq \bar{x}_\psi - 1;$$

and its goal function $(-x_\psi) \rightarrow$ lexmin. Find a solution x' of this problem. The variants are possible :

1) $x' \in \mathbf{Z^n}$, then process is finished;

2) $x' \notin \mathbf{Z^n}$, then the alternatives take place:

a) $\psi = 1$, then process is finished, otherwise $p = \psi$ and go to step 2;

b) $x'_\psi = \bar{x}_\psi + 1$, then go to step 1.

In L-class enumeration algorithms we obtain lexicographically increasing sequence of L-classes from set M/L representatives.

3 Decomposition algorithms

After fixing all variables z_i^r we obtain the transportation problem $T(z)$ from (1)-(6):

$$f_1(x,z) = \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij}x_{ij} \longrightarrow \min$$

s.t.

$$\sum_{j=1}^{m} x_{ij} \leq \sum_{r=1}^{w_i} a_i^r z_i^r,\ i = 1, \ldots, n,$$

and conditions (3),(6).

Denote $D(z)$ corresponding dual problem with variables $u_i \geq 0, v_j \geq 0, i = 1, \ldots, n; j = 1, \ldots, m$.

It is assumed that following unequality take place for the solvability of the (1)-(6):

$$\sum_{i=1}^{n} \max_r a_i^r \geq \sum_{j=1}^{m} b_j.$$

Algorithm.

Step 0. Formulate the initial integer programming problem $P^{(1)}$: to find lexicographically minimal integer solution to the system

$$\sum_{i=1}^{n}\sum_{r=1}^{w_i} a_i^r z_i^r \geq \sum_{j=1}^{m} b_j, \tag{7}$$

$$\sum_{r=1}^{w_i} z_i^r \leq 1, i = 1, \ldots, n. \tag{8}$$

Let $f^{(0)} = +\infty$ and go to step1 (iteration 1).

Denote an optimal solution of $P^{(k)}$ as $z^{(k)}$ and an optimal solution of $T(z^{(k)})$ as $x^{(k)}$.

Iteration $k (k \geq 1)$.

Step 1. Solve problem $P^{(k)}$ with the help of a L -class enumeration algorithm. If $P^{(k)}$ has no solution, then the solving process is finished: $z^{(q)}, x^{(q)}$ for which $f^{(k-1)} = f(z^{(q)}, x^{(q)})$

takes place is an optimal solution to the problem (1)-(6). Let $z^{(k)}$ be a solution to the $P^{(k)}$. Go to step2.

Step 2. Formulate and solve the transportation problem $T(z^{(k)})$. This problem has an optimal solution $x^{(k)}$, because (7) is valid for $z^{(k)}$. We obtain the values of the dual variables $u^{(k)}, v^{(k)}$, too. Calculate $f^{(k)} = \min\{f^{(k-1)}, f(x^{(k)}, z^{(k)})\}$. If $f(z^{(k)}, x^{(k)}) < f^{(k-1)}$, then $f^{(k)}$ substitutes for $f^{(k-1)}$ in the inequalities system of the problem $P^{(k)}$. Go to step3.

Step 3. Construct the following inequality (Benders cut):

$$\sum_{i=1}^{n}\sum_{r=1}^{w_i}(c_i^r - u_i^{(k)}a_i^r)z_i^r < f^{(k)} - \sum_{j=1}^{m} b_j v_j^{(k)}, \tag{9}$$

go to step 4.

Step 4. Formulate problem $P^{(k+1)}$: to find z, which is lexicographically minimal integer solution to the inequalities system of $P^{(k)}$ and (9). Go to step1 (to next iteration).

This method is finite and gives an optimal solution. The number of additional inequalities monotonously increases. We may input an upper bound for this number and modify the method keeping finiteness. Using this method we can obtain some approximate algorithms.

We have created computer system for analysis and solving of the IPP and obtained good experimental results with L-class enumeration and hybrid algorithms [5,6]. Regular cuts using in this algorithms has led to essential solving process acceleration. Experiment shown that L-coverings cardinality is closely connected to the efficiency of the solving of IPP by L-class enumeration algorithms and cutting plane ones. Now we develop special programs for the production-transportation problems.

References

[1] Bakhtin A.E., Kolokolov A.A., Korobkova Z.W. *Discrete production-transportation problems.* Novosibirsk, Nauka, 1978 (in Russian).

[2] Kolokolov A.A. *A study of integer programming problems and methods on the base of L-partition.* 33 Inter. Wiss. Koll. TH Ilmenau, 1988, pp.53–55 (in Russian).

[3] Kolokolov A.A. *Regular partitions in integer programming.* In: Methods for solving and analysis of the discrete optimization problems. Omsk, 1992, pp. 67-93 (in Russian).

[4] Kolokolov A.A. *On the L-structure of the integer linear programming problems.* Abstracts of the 16th IFIP conference on system modelling and optimization. P.1. Compiegne, France, 1993, pp.183-184.

[5] Zaikina G.M., Kolokolov A.A., *Experimental study in integer programming with application of L-partition. Discrete optimization and analysis of complex systems.* Novosibirsk, 1989, pp.26–56 (in Russian).

[6] Zaikina G.M., Kolokolov A.A., Levanova T.V., *Experimental comparison of some integer programming methods. Solving and analysis methods of discrete optimization problems.* Omsk, 1992, pp.25–41 (in Russian).

Optimal Control Methods for Logistics Queueing Networks

Warren B. Powell
Tassio A. Carvalho
Gregory A. Godfrey
Hugo P. Simao

Department of Civil Engineering
and Operations Research
Princeton University
Princeton, NJ 08544

Statistics and Operations Research
Technical Report SOR-94-8

May 27, 1994

Abstract

This paper introduces a new methodology for solving dynamic network flow problems - the Logistics Queueing Networks (LQN). A variety of problems in logistics involve the combined problem of moving freight from origin to destination while simultaneously managing the capacity required to move this freight. Standard formulations for real-world problems usually lead to intractably large linear programs. The LQN approach can take into account more real world detail and is considerably faster than classical LP formulations. The solutions generated using the LQN approach are shown to be within a few percentage points of the LP optimal solutions depending on the size of the capacity fleets.

1 Introduction

Traditionally, dynamic scheduling problems have been modelled as multi-commodity network flow problems. However, the level of detail required for real world applications result in impractically large linear programs. A new approach presented here is based on a discrete event dynamic system (DEDS), where demands are queued at terminals while waiting for available capacity. We call it a logistics queueing network (LQN). In contrast with classical queueing networks, which track the movement of customers through a series of servers, an LQN is characterized by the management of containers that handle the movements from one server to the next. When a container arrives at its destination, it becomes empty and available for its next movement. Since the flow of demands into a location might not equal the flow out of it, it is necessary to reposition containers empty from one location to another.

Companies that move freight have some degree of flexibility as to when a demand is shipped. Demands arise in a process very much like a Poisson process. Aside from utilizing resources optimally, there is a need to anticipate when and where demands will appear so that available equipment can be better positioned to satisfy customer needs.

Each demand can be precisely described by a list of attributes.

- *Call-in time* is the time when the transportation company is informed of the existence of a shipment to be moved from one terminal to another.
- *Earliest departure time (edt)* is the time when the shipment is first available at the origin terminal for pickup.
- *Due time* is the time by which the shipment must have been delivered to the destination terminal.
- *Latest departure time (ldt)* is the latest time the demand can leave the origin and get to the destination by the due time. The *start time window* is the time interval between the edt and the ldt. The movement can start at any time within this time window.
- *Revenue* is the return gained by transporting a shipment. If the demand cannot be satisfied, that is, if there is no equipment available at the origin to perform that movement, then the demand is considered lost and no revenue is received.

A comprehensive description of the attributes of demands and other classes of objects involved in road transportation can be found in Schrijver [5].

Capacity is the term that describes the equipment or collection of resources enabling a movement between two terminals. Capacity must be assigned to satisfy demands. There is no consolidation; that is, two units of demand cannot share the same unit of capacity at the same time.

At any moment there might be shipments available to be transported. Other shipments that have been called in will become available in the future. Using historical data, it is possible to forecast shipments that will be called in later. Therefore, as we plan to move available shipments, we must also anticipate moving shipments that will become available in the future.

In addition to satisfying customer demands, profits must also be maximized. Taking into account future events, a set of decisions assigning available capacity to available demands must be made for the near future. Capacity can be moved empty (relocated) from one terminal to another in order to satisfy demand waiting to be moved or to anticipate demands that have not yet appeared. However, there is a cost associated with relocating capacity. Therefore, the benefit from satisfying demands that would be lost otherwise must surpass the impact of moving capacity empty.

A discrete-time, dynamic network can be generated by replicating the spatial network at each time period (fig 1). Links represent the potential flow of capacity across time, either as inventory or as a move from one terminal to another. Only some of the possible links are represented in the figure. We adopt a planning horizon approach. The length of the planning horizon, T, is chosen to guarantee a high quality solution while keeping the problem at a reasonable size.

Solution techniques typically involve modelling the problem as a multi-commodity network flow problem. Crainic and Roy [1] developed the framework to satisfy stationary demands for an LTL carrier. This approach provides a good tool for planning purposes, but it is not effective for real time scheduling. Desrochers and Soumis [2] presented a set covering formulation for a crew scheduling problem where the trip schedule is known in advance. Linear programming models for this problem typically have a large number of columns, thus lending themselves to a column generation technique (see Jones *et al.* [4]).

After stating the assumptions to be used, the model for the linear relaxation of the problem is presented in the next section. The logistics queueing networks (LQN) approach is discussed in Section 3 and is compared to the linear relaxation in Section 4. Results and extensions are summarized in Section 5.

2 LP Formulation for the Linear Relaxation

Linear programming models for real world problems usually require simplifying assumptions to keep the problem size compatible with available technology.

For the problem of managing capacity, there is an underlying network structure that will be exploited to evaluate the solution quality of the assignment-based LQN network. To formulate this problem compactly, we make the following assumptions:

- There is only one type of capacity, and that type is compatible with all demands.
- There is no consolidation. Only one unit of demand can be transported by each unit of capacity at a time.
- Shipments from one terminal to another are transported in one time step. Intermediate stops are not considered.

This problem can be modeled as a network with side constraints. We define the following sets:

- $\mathcal{C}$ is the set of terminals i in the network.
- $\mathcal{D}$ is the set of demands d available within the planning horizon, T.
- $\mathcal{N}$ is the set of nodes $(i,t), i \in \mathcal{C}, t \leq T$, in the dynamic network.
- $\mathcal{L}$ is the set of dynamic links (i,j,t). The travel time between terminal i and terminal j is represented by $\tau_{i,j}$, thus the link (i,j,t) goes from node (i,t) to node $(j, t+\tau_{i,j})$.

- $\mathcal{T}_d$ is the set of feasible departure times for satisfying demand d, otherwise known as the *start time window*.
- $\mathcal{Q}_{i,j,t}$ is the set of demands d between i and j having t as a feasible departure time.
- $\mathcal{I}_{i,t}$ is the set of terminals j such that there is an inbound link (j,i,t).
- $\mathcal{J}_{i,t}$ is the set of terminals j such that there is an outbound link (i,j,t).

The decision variables are:

- $y_{d,t}$ is the flow of capacity covering demand d on the link (i,j,t), where i is the origin and j is the destination for demand d (fig 2).
- $w_{i,j,t}$ is the flow of capacity being repositioned empty along link (i,j,t). If $i = j$, $w_{i,i,t}$ represents the amount of capacity in inventory at terminal i from time t to time $t+1$.
- $x_{i,j,t}$ is the total flow of capacity on the dynamic link (i,j,t).

And the costs associated with the decision variables are:

- $r_{d,t}$ is the revenue generated by choosing to satisfy demand d at time t, i.e., the cost associated with $y_{d,t}$.
- $c_{i,j}$ is the cost of repositioning capacity over the link (i,j,t), i.e., the cost associated with $w_{i,j,t}$.

The amount of flow entering or leaving the network through each node (i,t) must also be specified. Typically, capacity is available at several terminals during the first time period. Capacity moving between two terminals during the first time period can be represented as flow into its destination at the time it arrives there. Capacity may also be added to or subtracted from specific nodes after the first time period for reasons such as maintenance. Let $f_{i,t}$ be the inflow(+)/outflow(-) of exogenous capacity at node (i,t).

This problem can be formulated as:

$$\max_{y,w} \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}_d} r_{d,t} y_{d,t} - \sum_{(i,j,t)\in\mathcal{L}} c_{i,j} w_{i,j,t} \tag{1}$$

subject to:

$$\sum_{t\in\mathcal{T}_d} y_{d,t} \leq 1 \quad \forall d \in \mathcal{D} \tag{2}$$

$$\sum_{d\in\mathcal{Q}_{i,j,t}} y_{d,t} + w_{i,j,t} - x_{i,j,t} = 0 \quad \forall (i,j,t) \in \mathcal{L} \tag{3}$$

$$\sum_{j\in\mathcal{J}_{i,t}} x_{i,j,t} - \sum_{j\in\mathcal{I}_{i,t}} x_{j,i,t-\tau_{j,i}} + w_{i,i,t} - w_{i,i,t-1} = f_{i,t} \quad \forall (i,t) \in \mathcal{N} \tag{4}$$

$$y_{d,t}\,,\ w_{i,j,t}\,,\ x_{i,j,t} \geq 0 \tag{5}$$

Constraints (2) limit the amount of capacity used to satisfy each unit of demand. Constraints (3) specify that all capacity flowing on a link must either satisfy demands or represent repositioning. Finally, flow conservation is enforced by (4).

While this formulation leads to an optimal solution, it is prone to degeneracy. Furthermore, branch and bound techniques may be necessary to provide integer solutions. The optimal objective function value for the linear relaxation provides an upper bound on the value using the LQN approach presented in the next section.

3 The Logistics Queueing Network Approach

Real time scheduling typically involves a decentralized decision process. At each terminal there is an individual—the dispatcher—responsible for assigning demands to capacity. Based on experience, he corrects the flow imbalance at his terminal by repositioning capacity. In this section we present the mathematical fundamentals behind the LQN approach that mimics these assignments.

3.1 Basic Idea

The key principle behind the LQN method is that, at any terminal and at any time, capacity has a value attached to it. The value of one unit of capacity at a given time is the sum of the value of the next activity for this capacity plus its future value at the destination at the expected time of arrival.

Load selection and capacity repositioning decisions directly affect the total profit of the network. A bi-level optimization strategy decouples these two types of decisions. The lower level optimization makes decisions about load selection, while the upper level sets upper bounds on the amount of capacity that can be repositioned over a link (i, j, t).

The notation defined in the previous section is also used here. Vectors are represented by stating only the indices common to every component. For example, $y_{t'}$ represents the vector of all $y_{d,t}$ with $t = t'$, and y represents the vector of all $y_{d,t}$. In addition, we define the following control variable:

- $u_{i,j,t}$ is the upper bound on the value of $w_{i,j,t}$.

The objective function for this upper bound problem can be stated as:

$$\max_u G(u) = \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}_d} r_{d,t} y_{d,t} - \sum_{(i,j,t)\in\mathcal{L}} c_{i,j} w_{i,j,t} \tag{6}$$

$G(u)$ represents the total profit resulting from all the activities, or movements, started within the time horizon. Computing $G(u)$ consists of choosing vectors y and w to maximize G for a given vector u. $G(u)$ also depends on the amount of capacity available at the first time period. Defining S_t to be the vector of capacity available at time t across all terminals, we can represent $G(u)$ by $F_1(S_1, u)$. We can compute $F_1(S_1, u)$ recursively. For notational compactness, we assume that all travel times are equal to one time period. The backward recursive formulation is then:

$$F_t(S_t, u) = \max_{y_t, w_t, S_{t+1}} \left(f_t(S_t, y_t, w_t, u_t) + F_{t+1}(S_{t+1}, u)\right) \tag{7}$$

for $t = 1, 2, \ldots, T$ where $f_t(S_t, y_t, w_t, u_t)$ represents the value of decisions made at time t,

$$f_t(S_t, y_t, w_t, u_t) = r_t y_t - c_t w_t \tag{8}$$

As the initial distribution of capacity is known, equation (7) can be used for running a forward pass simulation. In this case, F_{t+1} can be approximated by a linear function of the distribution of capacity at time $t+1$. The choice of S_{t+1} is not independent of y_t and w_t since it follows from the flow conservation constraints:

$$S_{k,t+1} = S_{k,t} - \sum_j x_{k,j,t} + \sum_i x_{i,k,t} \tag{9}$$

Once the decisions for time period t are made, the value of each component of S_{t+1} is known. Several constraints must be satisfied when maximizing F_t. First, only one unit of capacity may be assigned to each demand. Second, repositioning is restricted by the upper bounds u. Third, in order to carry out an activity, there must be available capacity. Also, the equations (9) and (3) must be satisfied. This results in the following linear program:

$$F_t(S_t, u) = \max_{y_t, w_t, S_{t+1}} (r_t y_t - c_t w_t + S_{t+1}\nu_{t+1}) \tag{10}$$

subject to:

$$\begin{aligned}
y_{d,t} &\leq 1 \quad \forall d \in \mathcal{Q}_{i,j,t} \ \forall i,j && (11)\\
w_{i,j,t} &\leq u_{i,j,t} \quad \forall i,j && (12)\\
\sum_{j\in\mathcal{J}_{i,t}} \sum_{d\in\mathcal{Q}_{i,j,t}} y_{d,t} + \sum_j w_{i,j,t} &\leq S_{i,t} \quad \forall i && (13)\\
S_{i,t+1} + \sum_j x_{i,j,t} - \sum_k x_{k,\acute{i},t} &= S_{i,t} \quad \forall i && (14)\\
\sum_{d\in\mathcal{Q}_{i,j,t}} y_{d,t} \ + w_{i,j,t} - x_{i,j,t} &= 0 \quad \forall (i,j,t) \in \mathcal{L} && (15)
\end{aligned}$$

This lower level optimization is then reduced to a series of subproblems for each terminal i. The use of the multipliers ν_t as the subgradient of $F_t(S_t)$ is clear from their role in equation (10). If $S_{i,t}$ increases by one unit, then $\nu_{i,t}$ reflects its impact in the objective function. However, the vector ν is not readily available, but it can be approximated using an iterative procedure.

3.2 Algorithm

We now show how to imbed the sequence of subproblems into an iterative procedure to update the vector of upper bounds u. The algorithm can be divided into three parts. The first part consists of the lower level optimization—find values for the vectors y_t and w_t by solving the linear program represented by equations (10)–(15) for each time period. The linear program decomposes into one independent subproblem for each terminal i.

The second part updates the multipliers ν. To approximate the multipliers more precisely, we distinguish between ν^+, the change in F resulting from adding one unit of capacity at a terminal, and ν^-, the change in F resulting from subtracting one unit of capacity at a terminal. That is,

$$\nu_t^+ = (\frac{\delta F_t(S_t)}{\delta S_t})_+ \tag{16}$$

$$\nu_t^- = (\frac{\delta F_t(S_t)}{\delta S_t})_- \tag{17}$$

Finally, the upper level optimization updates the vector of repositioning upper bounds, u, using the new multipliers ν. Iteratively, these three steps can reach a solution which is very close to optimal, as it is shown in the next section. A detailed description of each procedure employed in the algorithm follows.

Lower Level Optimization (Forward Pass) For each time period t, the subproblem can be solved if the capacity available at that time period, S_t, is known. Since only S_0 is known, we need to use a forward pass simulation starting at the first time period to create a feasible solution. If there is no consolidation, then an integer solution can be easily found by ranking the activities in the queue according to their revenue coeficients and greedily assigning capacity to the highest values.

Preliminary computational experiments show that using equation (10) as the subproblem objective function does not yield good results due to future uncertainty. While keeping the overall value function $G(u)$ as previously defined, discount factors are assigned to the cost components in the objective function (10) to discount future uncertainty.

The revenue coefficients in equation (10) to move demand d at time t are replaced by

$$V_{d,t} = r_{d,t} \max[1 - \beta\, l_t\, , 0] + \nu_{j,t+1}\, \alpha \tag{18}$$

where j is the destination of demand d and l_t is the difference between the latest departure time and t. The repositioning cost coefficients to move empty to terminal j are replaced by:

$$V_j = -c_{i,j} + \nu_{j,t+1}\, \alpha \tag{19}$$

The parameters β and α are calibrated using test runs. The value β represents the percentage by which the original demand revenue is discounted for each time period in the future that this demand is available. The value α weights the future value of being at terminal j into the revenue and cost coefficients. If $\beta = 0$ and $\alpha = 1$, then these cost coefficients are the same as in equation (10).

One iteration of the LQN algorithm considers the effects of marginal changes in the current solution. So, during the forward pass, we perform a sensitivity analysis to determine the effect of changing the total capacity by one unit at each node (i, t).

Multipliers update (Backward Pass) While assigning capacity to demands in the forward pass, we have to consider the local impact of gaining or losing one unit of capacity at each terminal at a given time. The value, $\nu_{i,t}$, of one unit of capacity at terminal i at time t is the sum of the contributions of all the activities of that unit of capacity from time t to the end of the planning horizon, T. It follows, then, that

$$\nu_{i,T+1} = 0 \quad \forall i \in C \tag{20}$$

The multipliers can then be computed recursively starting at time $t = T$ as explained below.

Computing ν^+: To compute $\nu_{i,t}^+$ one must consider the flow augmenting path from node (i, t). This computation uses the information gathered by the forward pass on any additional activity that would occur at time t and the status of that activity by the end of the forward pass. Three different cases may apply.

Case 1: The extra unit of capacity at (i, t) moves to terminal j a demand d that was lost at the end of the forward pass of the present iteration (fig 3).

$$\nu^{+}_{i,t} = r_{d,t} + \nu^{+}_{j,t+\tau_{i,j}} \tag{21}$$

Case 2: The extra unit of capacity at (i, t) immediately transports a demand that would have moved at a later time period t' during the present forward pass. Thus, the demand is satisfied in an earlier time period. We must then consider the flow augmenting path from (i, t).

If the flow augmenting path goes through the node $(j, t' + \tau_{i,j})$, then there is no significant change in the decision. The demand would be satisfied at an earlier time period and the unit of capacity originally assigned to it would be available at the origin i at time t'. Therefore, the multiplier is:

$$\nu^{+}_{i,t} = \nu^{+}_{i,t'} \tag{22}$$

If the flow augmenting path does not go through the node $(j, t'+\tau_{i,j})$, then the demand is satisfied in an earlier time period, but future decisions are affected as seen in figure 4. The subgradient can be computed by

$$\nu^{+}_{i,t} = \nu^{+}_{i,t'} + \nu^{+}_{j,t+\tau_{i,j}} + \nu^{-}_{j,t'+\tau_{i,j}} \tag{23}$$

Case 3: The extra unit of capacity at (i, t) is repositioned to terminal j.

$$\nu^{+}_{i,t} = -c_{i,j} + \nu^{+}_{j,t+\tau_{i,j}} \tag{24}$$

Computing ν^{-}: To compute $\nu^{-}_{i,t}$ one must consider the flow reducing path from node (i, t). This computation uses the information gathered by the forward pass on any activity that would not occur at time t if there was one less unit of capacity. Once again, three different cases exist.

Case 1: One less unit of capacity at node (i, t) loses the demand d.

$$\nu^{-}_{i,t} = -r_{d,t} + \nu^{-}_{j,t+\tau_{i,j}} \tag{25}$$

Case 2: One less unit of capacity at node (i, t) postpones the movement of the demand d from time t to a later time t'. If the flow reducing path goes though node $(j, t' + \tau_{i,j})$, then the multiplier is computed by

$$\nu^{-}_{i,t} = \nu^{-}_{i,t'} \tag{26}$$

If the flow reducing path does not go through node $(j, t' + \tau_{i,j})$, then the subgradient is computed by (fig 5)

$$\nu^{-}_{i,t} = \nu^{-}_{j,t+\tau_{i,j}} + \nu^{-}_{i,t'} + \nu^{+}_{j,t'+\tau_{i,j}} \tag{27}$$

Case 3: One less unit of capacity at node (i, t) cancels a repositioning move to terminal j.

$$\nu^{-}_{i,t} = c_{i,j} + \nu^{-}_{j,t+\tau_{i,j}} \tag{28}$$

Upper Level Optimization: Intuitively, an upper bound must increase or decrease when the impact of doing so increases the objective function G. The forward pass provides a feasible solution to the problem. The backward pass makes a comprehensive diagnosis of possible marginal changes in this feasible solution. The upper level optimization considers the change in the objective function by varying the upper bounds u. Besides the direct cost of the repositioning itself, the cost contribution associated with increasing an upper bound on a link (i, j, t) is the sum of the value of one less unit of capacity at i at time t and the value of one extra unit of capacity at j at time $t + \tau_{i,j}$. The impact of generating a repositioning over the link (i, j, t) is represented by the gradient

$$\Delta G^+_{i,j,t} = \nu^+_{j,t+\tau_{i,j}} + \nu^-_{i,t} - c_{i,j} \tag{29}$$

And the impact of eliminating one repositioning over the link (i, j, t) is

$$\Delta G^-_{i,j,t} = \nu^-_{j,t+\tau_{i,j}} + \nu^+_{i,t} + c_{i,j} \tag{30}$$

One needs to develop a policy to increase and decrease upper bounds based on the marginal value of these gradients. Due to network effects, one change in the decisions for the near future may result in several changes in the long-term decisions. The subgradients are estimated based on the marginal variation of only one component of the upper bound vector. Therefore, the performance of the algorithm is based on only one change per iteration, i.e., only one upper bound increases or decreases at each iteration.

4 Computational Experiments

In this section, the results obtained from the queueing approach are compared to the optimal solution of the linear relaxation model. The main purpose of the computational experiments is to compare the quality of the solutions obtained by the two methods. The logistics queueing network approach is also considerably faster than solving the LP using CPLEX.

The assumptions here are common to Sections 2 and 3. There is no consolidation. Each demand is moved from origin to destination in one step. There are no penalty costs related to keeping capacity in inventory. Capacity becomes available immediately after it reaches a terminal.

For the queueing approach, no repositioning is allowed in the first iteration, i.e., the vector of upper bounds u is initialized as a null vector.

Calibration: First, a suitable value for the parameters β and α in equations (18) and (19) must be found using a test data set. The set of terminals in the problem is a real set of terminals spread across the United States. A randomly generated set of demands was created for the calibration. Also, the initial distribution of capacity was randomly generated. The demand generation details are discussed in Godfrey [3].

The amount of capacity in the system has to be chosen so that the number of rejected demands reflect the rate at which demands are rejected in real world problems. In practice, this rate is close to 30% for truckload companies, but this rate is problem specific and also varies by season. We chose to evaluate two different capacity levels. The rejection rate was close to 35% for the first level and close to 20% in the second one.

The calibration data set involves 1295 demands distributed across five days. Time periods are two hours long. Shorter time periods are feasible for the queueing approach but due to degeneracy the solution times for the LP become too high.

Table 1 shows the percentage of the optimal value of the objective function reached using the queueing approach for the first level of capacity, which was 200 units of capacity in the system. Table 2 shows the same statistics for 250 units of capacity.

β	α				
	0.0	0.2	0.3	0.4	1.0
0.00	97.7	98.5	98.2	97.9	97.4
0.05	98.0	98.5	98.0	97.8	96.9
0.10	98.4	97.9	97.3	97.6	96.9
0.20	98.2	97.9	98.2	97.6	96.8

Table 1 : Percentage of optimal value - capacity = 200

β	α				
	0.0	0.2	0.3	0.4	1.0
0.00	99.4	99.6	99.5	99.2	99.0
0.05	99.1	99.7	99.7	99.4	99.2
0.10	99.1	99.6	99.4	99.7	99.6
0.20	98.6	99.7	99.3	99.6	96.2

Table 2 : Percentage of optimal value - capacity = 250

The parameter α weighs the value of one unit of capacity at the destination of an activity. In order to understand how α affects the solution one must look at two successive iterations. At any iteration, we rank demands, update subgradients, and adjust upper bounds. In the next iteration, demands are reranked using new subgradients. At any point in time, if $\alpha > 0$, then from one iteration to the next, the order in which demands are queued might change depending on the new subgradient at the destination of each demand. This change in priority was not anticipated in the backward pass of the previous iteration. For $\alpha = 0$, the subgradients do not affect the queueing of loads, and the objective function progresses in an almost monotonic fashion. For $\alpha = 1$, the subgradients shuffle the demand queues and often the sensitivity analysis of the backward pass in the previous iteration is inaccurate. The value $\alpha = 0.2$ is chosen to mitigate the shuffling of the demand queues while still using the subgradients to reach a better solution.

When queueing demands at any point in time, β represents the percentage by which the revenue generated by a load is discounted for each time period of availability in the future. However, as was mentioned previously, this discounting only applies for queueing purposes. If a demand is satisfied, its contribution to the objective function $G(u)$ is the same, regardless of the dispatching time. From tables 1 and 2, there is no clear trend on how the maximum value of the objective function varies with β, but $\beta = 0.05$ seems to yield consistently good results for $\alpha = 0.2$.

Numerical Results: Another six random demand sets generated from the same set of distributions were analyzed using the values of β and α previously selected. The percentage of the optimal solution found by the queueing approach for each demand set is shown in table 3.

data set	cap = 200	cap = 250
120.1	98.7	99.6
120.2	98.6	99.7
120.3	97.7	99.5
120.4	97.3	97.5
120.5	99.0	99.6
120.6	98.3	99.4

Table 3 : Percentage of optimal value for different data sets

For the lower capacity level, the results obtained here are comparable to those obtained from the first test. For the higher capacity level, the final results were clearly better. When capacity levels are restricted, problems are more tightly constrained so the solutions are further from optimality.

While the results were good in general, data set 120.4 presented problems for both capacity levels. Instead of initializing u as a null vector, a better way to improve upon the initial solution must be found to maintain a more consistent solution quality.

5 Conclusion

In the description of the algorithm, many simplifying assumptions were invoked only to make a consistent comparison with the LP model solution. Lifting some of these restrictions add very little complexity to the LQN algorithm:

- There can be multiple types of capacity. The units of capacity available at a terminal must be matched to demands. In this case the subproblem represented by equations (10)–(15) results in an assignment problem where each capacity-activity pair has a specified cost. The subgradients must represent the value of one more or one less unit of each type of capacity at each point in time.
- Demands can be routed through intermediate steps. Each step can be carried out by a different unit of capacity. In this case demands are queued at the intermediate terminals waiting for capacity to carry out the next step.
- Consolidation may be considered. However, the subproblem (10)–(15) turns out to be more complicated. Demands must be bundled together, resulting in a bin-packing problem.
- Instead of expiring after the due time, demands can have soft time-window constraints. Late demands have a penalty function assigned to them.
- After the subproblem is solved for each point in time, the cost components resulting from these decisions can be computed. Cost components related to holding costs and crews can be included easily.

Similar procedures to the LQN may be employed for a variety of problems. Problems where departures follow a fixed schedule can be successfully solved using this same strategy. Instead of considering discrete time and solving one assignment problem for each terminal at each time period, the assignment problem can be solved before every scheduled departure. In addition to providing a tool for real time scheduling, a series of strategical questions for different scenarios can be answered. For example, the effects of offering a new shipping frequency can be evaluated. If an increase in demand is forecasted, several hypothetical schedules can be compared for efficiency. The fleet size can be adjusted for strategic planning purposes. The method presented here also serves as a powerful simulation tool.

A new approach has been proposed for solving dynamic scheduling problems. This approach is flexible enough to include several real world details which would be overlooked in a straighforward linear programming formulation. Among these details, queuing networks provide much more flexibility than linear programs in composing cost functions.

The quality of the solution was evaluated by comparing with the LP optimal value of the objective function. Resource availability plays an important role in how close to optimality one can get using the queueing network approach. For loosely constrained problems, results were generally within 1% of the optimal solution. For the more tightly constrained problems, results were within 3% of optimality.

References

[1] T. G. Crainic and J. Roy, "Design of Regular Intercity Driver Routes for the LTL Motor Carrier Industry", *Transportation Science, v26.4, 1992.*

[2] M. Desrochers and F. Soumis, "A Column Generation Approach to the Urban Transit Crew Scheduling Problem", *Transportation Science, v23.1, 1989.*

[3] G. Godfrey and W. Powell, "Dynamic Programming for Spatial, Stochastic Scheduling Problems", *Technical Report SOR 94, Princeton University, 1994.*

[4] K. L. Jones *et al.*, "Multicommodity Network Flow The Impact of Formulation on Decomposition", *Technical Report SOR 91-23, Princeton University, 1992.*

[5] P. R. Schrijver, "Supporting Fleet Management by Mobile Communications", *Dissertation, The Netherlands, 1993 .*

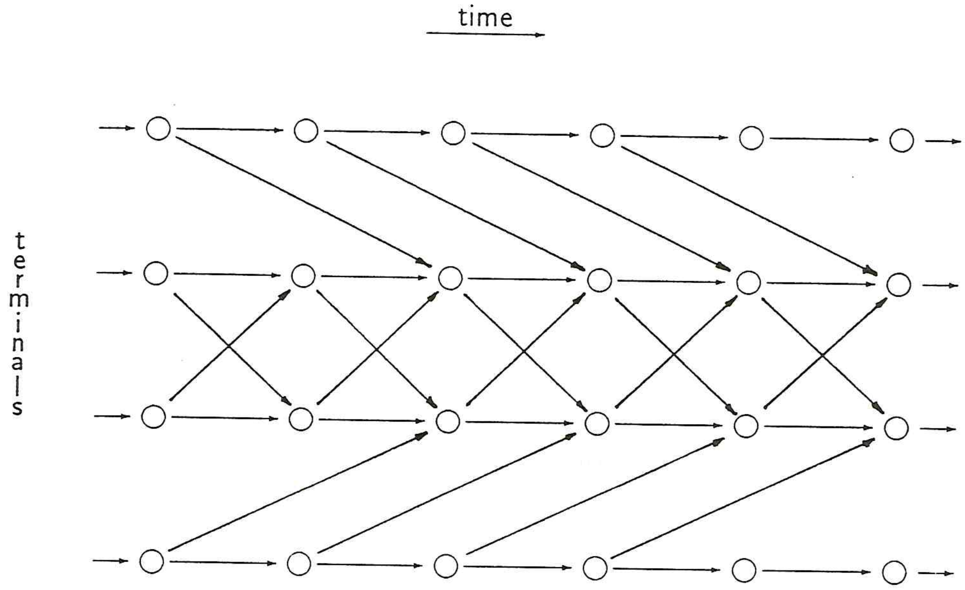


Figure 1: Dynamic Network

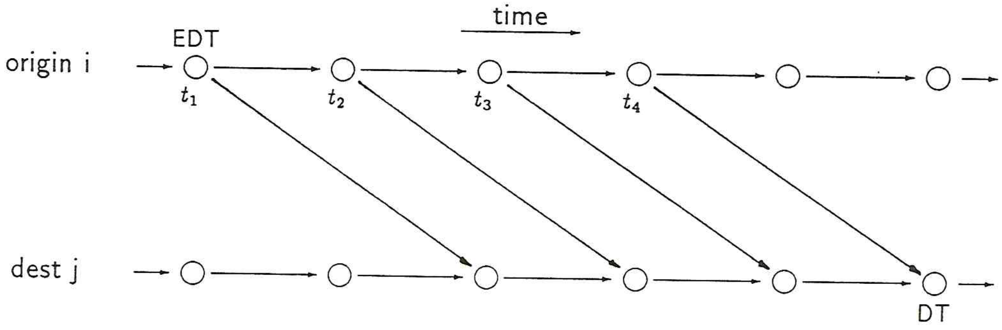


Figure 2: Feasible departure times for a demand

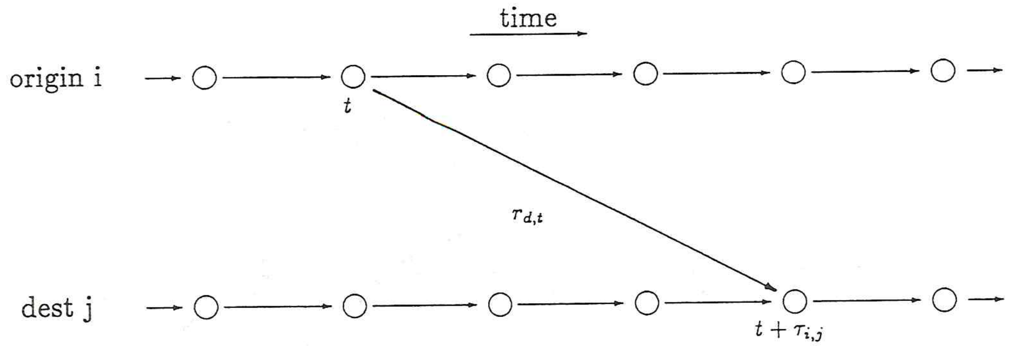


Figure 3: Case 1 for computing $\nu_{i,t}^{+}$

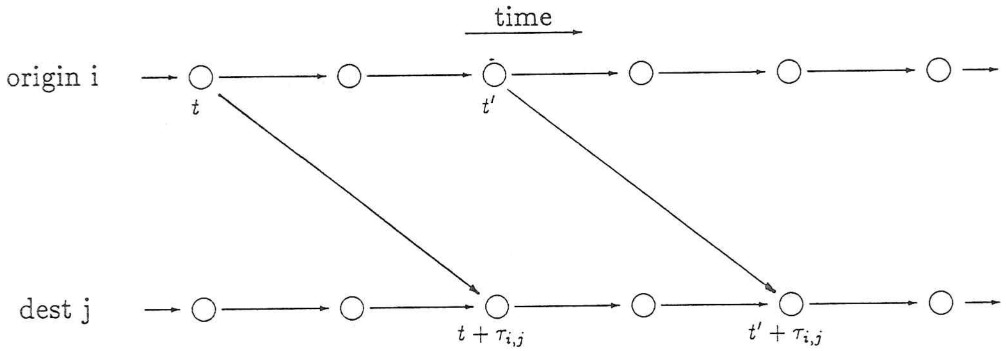


Figure 4: Case 2b for computing $\nu_{i,t}^{+}$

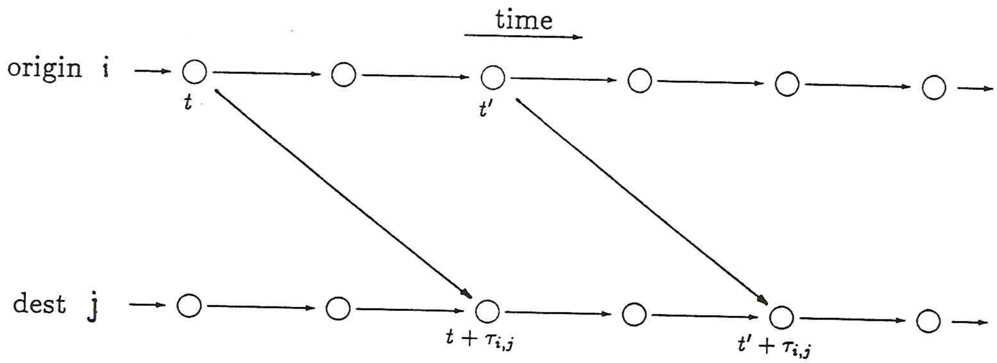


Figure 5: Case 2b for computing $\nu_{i,t}^-$

Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem

Julien Bramel
Graduate School of Business, Columbia University
David Simchi-Levi
Dept. of IE & MS, Northwestern University, and
Dept. of IE & OR, Columbia University

The Vehicle Routing Problem (VRP) can be stated as follows: a set of customers dispersed in a geographic region has to be served by a fleet of vehicles initially located at a given depot. Each customer has a load that must be picked up and the customer specifies a period of time, called a *time window*, in which this pick up must occur. The total load carried by each vehicle can be no more than the vehicle capacity. The objective is to find a set of routes for the vehicles, where each route begins and ends at the depot, serves a subset of the customers without violating the capacity and time window constraints, while minimizing the total length of the routes.

Due to the wide applicability and the economic importance of the problem in the service industry, variants of it have been extensively studied in the vehicle routing literature; for a review see Solomon and Desrosiers (1988). Most of the work has focused on *empirical analysis*, where heuristics are implemented on standard test problems and their performance is compared to other heuristics, see for example Solomon (1986) or Desrochers, Desrosiers and Solomon (1992). By contrast, very few papers have studied the problem from an analytical point of view in an attempt to characterize the theoretical behavior of heuristics.

To formally describe the model we analyze here, let the index set of the n customers be denoted $N = \{1, 2, \ldots, n\}$. Let $x_k \in \Re^2$ be the location of customer $k \in N$. Assume, without loss of generality, that the depot is at the origin and, by rescaling, that the vehicle capacity is 1 and that the length of the working day is 1. Associated with customer k is a quadruplet (w_k, e_k, s_k, l_k), called the customer *parameters*, which represents, respectively, the load that must be picked up, the earliest starting time for service, the time required to complete the service, called the *service time*, and the latest time service can end. Clearly, feasibility requires that $e_k + s_k \leq l_k$ and $w_k, e_k, l_k \in [0, 1]$, for each $k \in N$.

For any point $y \in \Re^2$, let $\|x\|$ denote the Euclidean distance between y and the depot. Let $d_k \equiv \|x_k\|$ be the distance between customer k and the depot. Also, let $d_{jk} \equiv \|x_j - x_k\|$ be the distance between customer j and customer k. Let Z_n^* be the total distance traveled in an optimal solution to the VRP, and let Z_n^{H} be the total distance traveled in the solution provided by a heuristic H.

Consider the customer locations to be distributed according to a distribution μ with compact support in $\Re^2$. Let the customer parameters $\{(w_k, e_k, s_k, l_k) : k \in N\}$ be drawn from a joint distribution Φ with a density ϕ. Let C be the support of ϕ, i.e., C is a subset of $\{(w, e, s, l) \in [0, 1]^4 : e + s \leq l\}$. Finally, we assume that a customer's location and its parameters are *independent* of each other.

1. The Asymptotic Optimal Solution Value

In the first part of the talk we concentrate on determining the asymptotic optimal solution value.

We associate a *job* with each customer. The parameters of job k are the parameters of customer k, that is, (w_k, e_k, s_k, l_k), where w_k is referred to as the *load* of job k and, using standard scheduling terminology, e_k represents the earliest time job k can begin processing, s_k represents the processing time, and l_k denotes the latest time the processing of the job can end. The value of e_k can be thought of as the *release time* of job k, that is, the time it is available for processing. The value of l_k represents the *due date* for the job. Occasionally, we will refer to customers and jobs interchangeably; this convenience should cause no confusion.

For a given set of customers $T \subseteq N$ with parameters $\{(w_k, e_k, s_k, l_k) : k \in T\}$, we associate a corresponding *machine scheduling problem* as follows: Consider the set of jobs T and an infinite sequence of *parallel* machines. Job k becomes available for processing at time e_k and must be finished processing by time l_k. The objective in this scheduling problem is to assign each job to a machine such that (i) each machine has at most one job being processed on it at a given time, (ii) the processing time of each job starts no earlier than its release time and ends no later than its due date, (iii) the total load of all jobs assigned to a machine is no more than 1, and the number of machines used is minimized. In our discussion we refer to (ii) as the *job time window constraint* and to (iii) as the *machine load constraint.*

Let $M^*(S)$ be the minimum number of machines needed to schedule a set S of jobs. It can be shown that if M_n^* is the minimum number of machines needed to schedule a set of n jobs whose parameters are drawn independently from a distribution Φ, then there exists a constant $\gamma > 0$ (depending only on Φ) such that $\lim_{n\to\infty} M_n^*/n = \gamma$ $(a.s)$.

Here we relate the solution to the VRP to the solution to the scheduling problem defined by the customers parameters. That is, we show that asymptotically the VRP is no more difficult to solve than the corresponding scheduling problem. Our first result is the following:

Theorem 1 *Let $x_1, x_2, \dots, x_n$ be independently and identically distributed according to a distribution μ with compact support in $\Re^2$, and let*

$$E(d) = \int_{\Re^2} \|x\| d\mu(x).$$

Let the customer parameters $\{(w_k, e_k, s_k, l_k) : k \in N\}$ be drawn independently from Φ. Let M_n^* be the minimum number of machines needed to feasibly schedule the n jobs corresponding to these parameters, and let $\lim_{n\to\infty} M_n^*/n = \gamma$ (a.s.). Then

$$\lim_{n\to\infty} \frac{1}{n} Z_n^* = 2\gamma E(d) \quad (a.s.).$$

In addition, we develop a heuristic, called the Location Based Heuristic, which has the property of being *asymptotically optimal.* This means that as the number of customers increases the heuristic solution's relative error with the optimal solution value goes to zero. This heuristic has also proven to be very effective on problems of moderate size, that is, the set of standard test problems of Solomon (1986).

2. Analysis of the Set Partitioning/Column Generation Technique

A classical method used to solve the VRP is based on formulating the VRP as a Set Partitioning Problem. Let the index set of all feasible routes be $\{1, 2, \ldots, R\}$ and let c_r be the length of route r. Define

$$\alpha_{ir} = \begin{cases} 1, & \text{if customer } i \text{ is served in route } r, \\ 0, & \text{otherwise,} \end{cases}$$

for each $i = 1, 2, \ldots, n$ and $r = 1, 2, \ldots, R$. Finally, for $r = 1, 2, \ldots, R$, let

$$y_r = \begin{cases} 1, & \text{if route } r \text{ is in the optimal solution} \\ 0, & \text{otherwise.} \end{cases}$$

In the *Set Partitioning* formulation of the VRP, the objective is to select a minimum cost set of feasible routes such that each customer is included in some route. It is the following integer program:

$$\begin{aligned} \text{Problem } P: \quad & Min \sum_{r=1}^{R} y_r c_r \\ s.t. \quad & \\ & \sum_{r=1}^{R} y_r \alpha_{ir} \geq 1, \quad \forall i = 1, 2, \ldots, n \qquad (1) \\ & y_r \in \{0, 1\}, \quad \forall r = 1, 2, \ldots, R. \end{aligned}$$

This formulation was first used successfully by Cullen, Jarvis and Ratliff (1981) to design heuristic methods for the VRP. Recently, Desrochers, Desrosiers and Solomon (1992) have used it in conjunction with other methods to generate optimal or near optimal solutions to the VRP.

Of course, the set of all feasible routes is extremely large and one cannot expect to generate it completely. Even if this set is given, it is not clear how to solve the set partitioning problem since it is a large scale integer program. To overcome the first difficulty, Desrochers, Desrosiers and Solomon use the celebrated column generation technique, which makes it possible to solve the linear programming relaxation of Problem P without having to enumerate all the routes. This is done by enumerating a portion of all possible routes, and solving the resulting linear programming relaxation with this partial route set. The solution to the linear program is then used to determine if there are any routes not included which can reduce the solution value. This is the *column generation* step. Using the values of the optimal dual variables (with respect to the partial route set), we generate a new route and resolve the linear programming relaxation of the set partitioning problem. This is continued until one can show that an optimal solution to the linear program is found; one that is optimal for the complete route set. Finally, to get an integer solution to the set partitioning problem, the linear program is combined in a branch and bound routine.

It is well known that a branch and bound strategy works well only if the lower bound used in the bounding step is very tight. Fortunately, many researchers have reported that the linear program relaxation of the set partitioning problem provides a solution close to the optimal integer solution, see e.g. Desrochers, Desrosiers and Solomon. That is, the solution to the linear programming relaxation of Problem P provides a very tight lower bound to the solution of the VRP. In their paper, Desrochers, Desrosiers and Solomon report an average relative gap between the optimal solution to the linear programming relaxation and the optimal integer solution of only 0.733%. In addition, if one looks more closely at their results, the average relative gap on the problems with 25 customers is 1.123%, on the problems with 50 customers it is 0.276%, and on the problems with 100 customers it is 0.028%. That is, the average relative gap, at least in their computational study, decreases as the number of customers increases. Here we demonstrate why this is true in general.

For this purpose, we perform a probabilistic analysis of the linear programming relaxation of Problem P. Our second result is the following:

Theorem 2 *Let the customer locations be independently and identically distributed according to a distribution μ with compact support in $\mathbb{R}^2$. Let the customer parameters be independently and identically distributed like Φ with support in $[0,1]^4$. Let Z^{LP} be the value of the optimal fractional solution to P, and let Z^* be the value of the optimal integer solution to P; that is, the value of the optimal solution to the VRP. Then,*

$$\lim_{n\to\infty} \frac{1}{n} Z^{\mathrm{LP}} = \lim_{n\to\infty} \frac{1}{n} Z^* = 2\gamma E(d) \quad (a.s.).$$

Therefore, the relative gap between the value of the solution to the linear programming relaxation of P and the optimal integer solution to P (a solution to the VRP) goes to zero as the number of customers increases.

An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands

by

Gilbert Laporte, Centre de recherche sur les transports
Université de Montréal, Case postale 6128, succursale A
Montréal, Canada H3C 3J7

François Louveaux, Facultés Universitaire Notre-Dame de la Paix
8, Rempart de la Vierge, B-5000 Namur, Belgium

Luc Van hamme, Beyers Innovative Software
Michielssendreef 40/42, B-2130 Brasschaat, Belgium

1. Introduction

The classical deterministic *Vehicle Routing Problem* (VRP) can be defined as follows. Let $G = (V, E)$ be an undirected graph where $V = \{v_1, ..., v_n\}$ is a set of vertices representing cities or customers, and $E = \{(v_i, v_j) : i < j; v_i, v_j \in V\}$ is an edge set. With each vertex $v_i (i \geq 2)$ is associated a non-negative demand q_i. With each edge (v_i, v_j) is associated a non-negative cost (distance, travel time) c_{ij}. Vertex v_1 represents a depot at which are based m identical vehicle of capacity $Q > 0$. Depending on the version of the problem considered, the value of m is either fixed, or bounded above by a constant $\overline{m}$. The VRP consists of determining vehicle routes in such a way that (i) all routes start and end at the depot; (ii) each vertex other than the depot is visited exactly once; (iii) the total demand of any given route does not exceed Q; (iv) the total distance traveled by all vehicles is minimized.

In the *Stochastic Vehicle Routing Problem* (SVRP), the demand associated with vertex v_i is a random variable ξ_i. As a result, it is no longer possible to assume that vehicle routes may be followed as planned as the total accumulated demand along a route may at some point exceed the vehicle capacity. The SRVP is modeled in two stages. In the first stage, *a priori* vehicle routes satisfying conditions (i) and (ii) are constructed, without full information on the demands. In the second stage, when this information becomes available, routes are followed as planned, until the accumulated demand attains or exceeds the vehicle capacity. In this case, *failure* is said to occur and a *recourse* action is taken: the vehicle returns to the depot to unload, and resumes its visits at the point of failure. The SVRP consists of determining an *a priori* set of routes so as to minimize the expected cost of the second stage solution. The SVRP is considerably more difficult to solve to optimality than its deterministic counterpart. For recent references on this problem, see Laporte, Louveaux and Mercure (1989), Dror, Laporte and Trudeau (1989), Laporte and Louveaux (1990), Bertsimas (1992), Gendreau, Laporte and Séguin (1992), and Dror, Laporte and Louveaux (1993).

2. Model

The SVRP can be formulated as a stochastic integer program as follows. Define integer first stage decision variables x_{ij} equal to 1 if $1 \leq i < j \leq n$ and edge (v_i, v_j) is used for a single trip in the first stage solution, to 2 if $i = 1, j > 1$, and edge (v_1, v_j) is used for a return trip, and to 0 otherwise. In what follows, c_{ij} and x_{ij} must be interpreted as c_{ji} and x_{ji} whenever $i > j$. Let $T(x, \xi)$ be the cost of the second stage solution if $x = (x_{ij})$ is the first stage solution, and $\xi = (\xi_i)$ is the vector of non-negative random variables associated with the vertices of $V \setminus \{v_1\}$. The SVRP formulation is then

$$\min_x \; E_\xi T(x, \xi) \tag{1}$$

subject to

$$\sum_{j=2}^{n} x_{1j} = 2m \tag{2}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \quad (v_k \in V \setminus \{v_1\}) \tag{3}$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - 1 \quad (S \subset V \setminus \{v_1\}, 3 \leq |S| \leq n-2) \tag{4}$$

$$0 \leq x_{1j} \leq 2 \qquad (v_j \in V \setminus \{v_1\}) \tag{5}$$

$$0 \leq x_{ij} \leq 1 \qquad (1 < i < j \leq n) \tag{6}$$

$$1 \leq m \leq \overline{m} \tag{7}$$

$$x_{ij} \text{ integer} \qquad (1 \leq i < j \leq n). \tag{8}$$

In this formulation, the constraints are those of the classical *m-Traveling Salesman Problem.* The objective can be recast as a two-stage program as follows. The length of the *a priori* solution is equal to cx. For a particular realization ξ of the random demand, the cost of recourse is $Q(x,\xi)$. Letting $\mathcal{Q}(x) = E_\xi Q(x,\xi)$, the objective can be rewritten as

$$\min_x \; cx + \mathcal{Q}(x). \tag{9}$$

3. Solution approach

The model defined by (2) – (9) can be solved using the following relaxation approach proposed by Laporte and Louveaux (1993). Define the initial current problem as

$$\min_{x,\theta} \; cx + \theta \tag{10}$$

subject to (2) – (3), $0 \leq x \leq 1$, and $\theta \geq L$. With respect to the initial SVRP statement, this formulation contains three types of relaxation : integrality and subtour elimination constraints have been omitted, and the term $\mathcal{Q}(x)$ has been replaced by an approximation θ, with a lower bound L. As is standard in several TSP algorithms, integrality is regained through branch and bound, and violated subtour elimination constraints are introduced where necessary. Similarly, the *recourse function* $\mathcal{Q}(x)$ is gradually approximated through the introduction of *optimality cuts* to be described below. The algorithm can be summarized as follows.

Step 0. Set ν, the current iterate point, equal to 0, and set $\overline{z}$ equal to the cost of a feasible solution. The only pendant node of the search tree corresponds to the initial current problem.

Step 1. Select a pendant node from the list. If none exists, stop.

Step 2. Set $\nu := \nu + 1$. Let (x^ν, θ^ν) be an optimal solution to the current problem.

Step 3. If $cx^\nu + \theta^\nu > \overline{z}$ fathom the current problem and return to Step 1. Otherwise, check for any violated subtour elimination constraint (4). If one can be identified, augment the current problem accordingly, and return to Step 2.

Step 4. Check for integrality restrictions. If one is violated, create new branches following the usual rules; append the new nodes to the list of pendant nodes; return to Step 1.

Step 5. Compute $\mathcal{Q}(x^\nu)$ as in Bertsimas (1992) or as in Gendreau, Laporte and Séguin (1992). Set $z^\nu = cx^\nu + \mathcal{Q}(x^\nu)$. If $z^\nu < \overline{z}$, set $\overline{z} := z^\nu$.

Step 6. If $\theta^\nu \geq Q(x^\nu)$, fathom the current node and return to Step 1. Otherwise impose one optimality cut and return to Step 2. ■

The optimality cuts imposed in Step 6 are derived as follows. Assume the feasible solutions are indexed by ν, and let $E_\nu = \{(v_i, v_j) : v_i, v_j \in V \setminus \{v_1\}$ and (v_i, v_j) belongs to solution $\nu\}$. Then the optimality cuts are

$$\theta \geq \frac{1}{2}(Q(x^\nu) - L)\left(\sum_{(v_i,v_j)\in E^\nu} x_{ij} - n + m\right) + Q(x^\nu) \qquad (11)$$

(see Laporte, Louveaux and Mercure (1992) and Laporte and Louveaux (1993)).

4. Implementation

The cuts defined by (11) guarantee that the algorithm terminates in a finite number of operations. To speed up convergence, they must be used in conjunction with other valid constraints. One possibility is to generate other types of valid constraints in Step 3, such as 2–matching constraints or comb inequalities. A key ingredient of the proposed algorithm is the derivation of a lower bounding functional for $Q(x)$. These developments are currently under way and will now be sketched briefly.

The optimality cuts defined by (11) can be improved by examining bounds on the value of the recourse function associated with neighbors of the current solution. Generic expressions for these improved cuts are provided by Propositions 6 and 7 of Laporte and Louveaux (1993). In a number of SVRPs, strong cuts can be imposed. An interesting case is the SVRP where at most one failure per route can occur. This case is realistic as it makes little managerial sense to plan vehicle routes which systematically fail. In this context, one idea is to generate cuts by examining neighbor solutions obtained by switching two consecutive vertices of a vehicle route

Another idea is to derive lower bounds on the cost of recourse by considering "clusters" of vertices linked by integer edges in the current solution. Indeed, the probability of having a failure within a cluster can be computed and this can be used to derive the desired lower bound. This idea can be extended by considering clusters of vertices linked by positive (possibly fractional) edges. For this, "connected components" must be identified as is commonly done for the TSP (see, e.g., Padberg and Rinaldi, 1990).

Testing these ideas requires a fair amount of fine tuning and experimentation and we will report on this at TRISTAN II. Preliminary results indicate that medium size instances can be solved to optimality using the proposed approach.

5. References

1. Bertsimas, D.J., "A Vehicle Routing Problem with Stochastic Demand", *Operations Research* 40 (1992), 574–585.
2. Dror, M., Laporte, G., Louveaux, F.V., "Vehicle Routing with Stochastic Demands and Restricted Failures", *Zeitschrift für Operations Research* 37 (1993), 273–283.
3. Dror, M., Laporte, G., Trudeau, P., "Vehicle Routing with Stochastic Demands : Properties and Solution Frameworks", *Transportation Science* 23 (1989), 166–176.
4. Gendreau, M., Laporte, G., Séguin, R., "The Vehicle Routing Problem with Stochastic Customers and Demands", Publication 873, Centre de recherche sur les transports, Montreal, 1992.
5. Laporte, G., Louveaux, F.V., "Formulations and Bounds for the Stochastic Capacitated Vehicle Routing with Uncertain Supplies", in : *Economic Decision-Making : Games, Econometrics and Optimisation*, J.J. Gabszewicz, J.-F. Richard and L.A. Wolsey (eds.), North-Holland, Amsterdam, 1990, 441–455.
6. Laporte, G., Louveaux, F.V., "The Integer L-Shaped Method for Stochastic Integer Programs with Complete Recourse", *Operations Research Letters* 13 (1993), 133–142.
7. Laporte, G., Louveaux, F.V., Mercure, H., "The Vehicle Routing Problem with Stochastic Travel Times", *Transportation Science* 26 (1992), 161–170.
8. Laporte, G., Louveaux, F.V., Mercure, H., "Models and Exact Solutions for a Class of Stochastic Location-Routing Problems", *European Journal of Operational Research* 39 (1989), 71–78.
9. Padberg, M.W., Rinaldi, G., "Facet Identification for the Symmetric Traveling Salesman Problem", *Mathematical Programming* 47 (1990), 219–257.

THE VEHICLE ROUTING PROBLEM WITH STOCHASTIC CUSTOMERS AND DEMANDS

by

René Séguin [1,2]

Michel Gendreau [1,3]

Gilbert Laporte [1,3]

[1] Centre de recherche sur les transports

[2] Université Libre de Bruxelles

[3] Université de Montreal

The classical deterministic *Capacitated Vehicule Routing Problem* (CVRP) is defined as follows. Let $G = (V,E)$ be a graph where $V = \{v_1,...,v_n\}$ is the vertex set and $E = \{(v_i,v_j)\}$ is the edge set. Vertex v_1 represents the *depot* while the remaining vertices correspond to *customers* or *cities*. With each vertex v_i is associated a non-negative demand d_i. We assume the graph is symmetrical, so that (v_i,v_j) is only defined for $i < j$. With each edge (v_i,v_j) is associated a non-negative *cost* or *distance* c_{ij}. It is assumed that matrix (c_{ij}) satisfies the triangle inequality. There is a fleet of m vehicles based at the depot. These vehicles are usually identical and of capacity D. Depending on the version of the problem under consideration, m is either a fixed constant, or a decision variable. Vehicles makes *collections* or *deliveries* , but not both. For convenience, we only consider collection routes. The CVRP consists of determining a set of m vehicle routes of minimal total cost so that 1) each route starts and ends at the depot ; 2) each customer is visited exactly once by one vehicle ; 3) the total customer demand on any route does not exceed D.

The CVRP is a well-known NP-hard problem and considerable effort has been devoted to its solution. The best available exact algorithms use dynamic programming coupled with state space relaxation, or enumerative methods based on the identification of "q-routes". Using these methods, the largest problems that can be solved to optimality contain approximately 50 vertices. Several heuristics have also been proposed for the CVRP. Currently, the most promising methods are based on *tabu search.*

In many practical contexts, one or several parameters of the CVRP are stochastic. This has a major impact both on how the problem is formulated and solved. Relatively little effort has been devoted to the study of stochastic CVRPs, as opposed to their deterministic counterparts. These problems are generally regarded as computationally intractable. There is hardly any usable theory for stochastic integer problems and almost no attempt has ever been made to solve these problems to optimality.

In this work, we study a class of CVRPs with stochastic demands and customers denoted by the abbreviation VRPSDC. This problem is also defined on a graph $G = (V, E)$. Each vertex has a probability p_i of being present and has a stochastic demand e_i; when vertex v_i is absent, its demand e_i is equal to zero. We assume all demands are discrete and independent. In a *first stage*, a set of routes satisfying conditions 1) and 2) of the CVRP are determined. Note that in the first stage problem capacity constraints are never considered resulting in a much larger solution space than in the deterministic case (CVRP). The set of vertices with zero demand is revealed before routing actually starts, but the positive demand of every remaining customer becomes known only when the vehicle arrives at the customer's location. In a *second stage*, the first stage routes are followed as planned, with the following two exceptions: 1) any absent customer is skipped; 2) whenever the vehicle capacity becomes exceeded, it returns to the depot to unload, and resumes collections starting at the last visited customer; if for any customer the vehicle capacity becomes exactly attained, the vehicle then returns to the depot and resumes collections at the next present customer along its route; in either case, *route failure* is said to occur. The VRPSDC consists of designing a first stage solution so as to minimize the expected cost of the second stage solution.

The VRPSDC belongs to the class of "*a priori* optimization problems", and to the family of stochastic integer programs with complete recourse. It generalizes two classes of problems that have been treated separately by a number of authors: the *Probabilistic Traveling Salesman Problem* (PTSP) and the *VRP with Stochastic Demands* (VRPSD). *A priori* optimization is a natural solution approach in contexts where it is impractical to recompute an optimal solution whenever the random variables are revealed. This can be because computing a new optimal solution at short notice is impossible. Another reason particular to vehicle routing is the importance for drivers to become acquainted with their route and regular customers. It then makes sense to work from a planned route, and to deviate from it as little as possible. In this sense, the objective of *a priori* optimization is similar to the one used in fixed routes problems. However, the way to handle those problems differs totaly from our.

In the PTSP there is just one vehicle, no demands and the only stochastic component of the problem is the subset of customers requiring a visit. The main results for the PTSP relate to the combinatorial structure of the problem but a number of asymptotic results have been derived and heuristics for the one vehicle and for the multi-vehicle versions of the problem have also been investigated.

The VRPSD, in which all customers are present but have random demands, has been more widely studied. Applications related to the delivery of home heating oil, to sludge disposal, where sludge accumulation at a plant is a random process, and to cash collection from bank branches are documented examples. A number of authors have studied the "chance constrained" version of the VRPSD, i.e., problems in which the probability of route failure is controlled but the cost of failure is not accounted for. Heuristic strategies have been proposed for this case and it has been shown that under some circumstances the chance constrained VRPSD can be tranformed into an equivalent deterministic VRP. The "recourse" version of the VRPSD consists, like for the VRPSDC studied in our work, of constructing vehicle routes of minimum expected cost. Properties and formulations of the VRPSD with recourse have been investigated by a number of authors and a savings heuristic algorithm has been proposed for this case.

Our version of the CVRP, with stochastic customers and demands, is more realistic as it combines two levels of uncertainty that coexist in several contexts. For example, less-than-truckload carriers often make collections at a set of regular customers on a periodic basis, e.g., daily, but not all customers require the vehicle's visit every day and this can be known just before starting collections; these customers are simply dropped from the planned route. The quantity to be collected at every remaining customer is a random variable known upon arrival at the customer's location and it may happen, at some point on the collection route, that the vehicle capacity becomes attained or exceeded. In such a case, a trip to and from the depot is necessary. The VRPSDC has received attention only recently and only a few papers have appeared on the subject; some theoretical properties have been investigated and a heuristic has been proposed and its asymptotic performance studied.

One of the obvious results from the previous studies is that the realism gained by introducing stochastic elements in TSPs or VRPs has increased the complexity of already difficult problems. It has been shown that optimal solutions to deterministic problems can be very poor solutions to their stochastic counterparts. Furthermore, the various properties and solution techniques used in the deterministic case are usually inadequate in a stochastic context and this forces one to be very careful when using deterministic results. In fact, new solution procedures, formulations and properties have to be derived for almost every specific stochastic VRP .

The main difficulties that we have to face with the VRPSDC are the presence of a non-linear objective function, of integer variables and of an exponentially large number of constraints. We formaly define the problem in the context of stochastic programming and provide a two-stage non-linear stochastic integer program with recourse. To solve the problem, we propose using the integer L-shaped method which works with a relaxation of the objective function and of some of the constraints. A branch and cut scheme is applied to regain feasibility and to gradually approach the original objective from below. This procedure extends the classical L-shaped method for continous variables to stochastic programs with first stage binary (integer) decision variables. It has already been applied with some degree of success to a number of combinatorial optimization problems and can be implemented by making suitable modifications to existing branch and bound solvers for mixed integer programs that allow for easy generation of additional constraints. Three different implementations of the method have been used and analyzed.

Previous articles on the m-TSP and on deterministic CVRPs have shown that problem difficulty is directly related to size (n), expected filling (f) and, to a lesser extent, to the number of vehicles (m). We have tested those factors on several series of randomly generated problems. In addition, we have carried out numerical experiments on "problem type", i.e. whether customers and demands were deterministic or stochastic. Also, the effects of the number s of stochastic customers and of the interval $[a_i,b_i]$ used to generate the probabilities p_i were analyzed.

Our results show that of the two stochastic components of the problem, stochastic customers are more difficult to handle than stochastic demands. Depending on parameter values, fully stochastic problems can be solved exactly for small to medium sizes. Problems with only stochastic demands are easier and exact solutions for this case can be obtained for larger problems. Until now, VRPs of the type studied in this work have always been considered as intractable and we therefore view our results as encouraging.

Probabilistic Analysis of a Combined Partitioning and Math Programming Heuristic for a General Class of Vehicle Routing and Scheduling Problems

Awi Federgruen Garrett van Ryzin [*][†]

December 9, 1993

Introduction and Motivation

Region partitioning schemes, in which solutions for a large service region are generated by combining solutions formed on smaller subregions, have always played a prominent role in the probabilistic analysis of Euclidean vehicle routing problems. The seminal paper of Beardwood, Halton and Hammersley [2] for the TSP, Karp's algorithm [9] and Steele's [10] general theory of subadditive, Euclidean functionals all employ variations of region partitioning schemes. For the capacitated vehicle routing problem (VRP), region partitioning schemes were exploited in the pioneering work Haimovich and Rinnooy Kan [8], and they form the foundation for much of the work that has followed since then (see Federgruen and Simchi-Levi [4] for a review). For example, the location-based heuristic of Bramel and Simchi-Levi [3] is analyzed by bounding its solution cost with that of a solution produced by a region partitioning scheme. Federgruen and van Ryzin [5] also use a partitioning scheme, together with a math-programming-based heuristic for general bin packing problems, to solve a capacitated vehicle routing problem with time window constraints.

The appeal of region partitioning schemes lies in their ability to approximate the solution structure of certain classes of VRPs. The service region is usually split into many smaller subregions, and only customers within one subregion are clustered when constructing tours. In this way, customers on a given tour are tightly grouped, and this allows one to bound travel costs. For example, the total distance of a tour serving a given subregion is never less than twice the distance from the depot to the closest point in that subregion, and the travel distance between two customers in a subregion in no more than the subregion's diameter, etc.. The resulting tour structure has tightly clustered collections of customers connected

[*] Graduate School of Business, Columbia University, NY, NY, 10027.

[†] The research of both authors was supported by an internal grant from the Columbia Graduate School of Business

to the depot by radial arcs. The problem is then analyzed by letting the grid size tend to zero as the size of the problem instance grows. Under certain probabilistic assumptions, one can show that with high probability the resulting limiting tour structure is asymptotically optimal (see [4]).

However, while such a solution structure is intuitively appealing, it need not be optimal in general. The optimality of this structure is in reality a consequence of the underlying probabilistic model that describes how problem instances are generated. More precisely, a typical assumption (see [4]) in probabilistic analysis is that the location of a customer is independent of its other attributes, such as delivery requirements, time windows, etc. Thus, in large instances, it is highly probable that a good collection of customers to combine into a tour can be found within close proximity of one another; that is, within a given neighborhood, one can expect to find roughly the same range of customer types in the same proportions as would be found anywhere else in the service region. Thus, for large instances it makes little sense to consider customers outside small neighborhoods. it is precisely for this reason that the solution structure mentioned above is close to optimal.

Nevertheless, one can show that there are many problems for which such a structure is quite far from optimal (even asymptotically) and for which region partitioning schemes – and algorithms relying on region partitioning for their analysis – can perform badly. For example, suppose that there are two disjont subsets of a service region, A and B, that are close together but far from the depot. Customers in A require a pick-up of a full load (loads travel from customer locations to the depot) and those in B require delivery of a full load (loads travel from the depot to customer locations). A region partitioning scheme superimposes a fine grid on the service region and forms tours that consist only of customers within a single subsquare of the grid. For a sufficiently small grid, this results in tours that service only one type of customer (either all A's or all B's). In this case, it is clearly better to combine a delivery to B with a pick-up from A rather than to make deliveries and pick-ups using separate tours. In this case, one can show that a paritioning heuristic constructs a solution that is almost twice the cost of the optimal solution.

In practice, it is likely that many problem instances would exhibit dependencies between locations and attributes such as load size, delivery time requirements, etc.. Does this mean that the long tradition of using partitioning ideas must be abandoned if more realistic models of problem instances are to be addressed? The answer, as we show below, is most definitely – No. Our analysis shows that far from being obsolete, the idea of using partitioning has in fact not been extended far enough.

A Combined Partitioning and Math Programming Heuristic

Our central idea is to apply partitioning schemes not only to customer locations, but also to their various attribute values, such as delivery requirements, scheduling constraints, etc. Using this *combined* partitioning of customers, we are able to generate heuristics for a wide class of vehicle routing and scheduling problems that are provably close to optimal for very general distributions of problem instances.

Our heuristic uses the partitioning of customer attributes and locations to aggregate customers into a finite number of types. A math program is solved based on these aggregated customer types to generate a feasible solution to the original problem. The problem class we address is quite general; feasible tours are defined by a given collection of ordered sets of customer types which need only satisfy a number of general consistency properties. The restrictions allowed on tours can involve general distance norms and a myriad of practical constraints, including vehicle capacity limits, time window restrictions and combined pick-up and delivery requirements. Associated with each tour is a cost, determined by a function again merely satisfying some general conditions. This class includes objectives that minimize total distance under any norm, number of vehicles and also inventory/routing costs.

We provide a probabilistic analysis of this heuristic under very general probabilistic assumptions. In particular, we do not require independence between customer locations and their various attribute values. Thus, our analysis is valid for problem instances that violate the independence assumptions required for convergence of most region partitioning heuristics. The results are obtained by constructing bounds on the corresponding mathematical programs. Using these bounds, we show that, our heuristic is (a.s.) ϵ-optimal as the number of customers n tends to infinity. Further, it runs in $O(n \log n)$ time for a fixed relative error, and can be designed to be asymptotically optimal while still running in polynomial time. We characterize the asymptotic average value of the heuristic and the optimal solution as the limit of a sequence of linear program values. We also provide bounds on the rate of convergence to the asymptotic value and bounds on tail probabilities.

The heuristic is an extension of the math programming heuristic proposed in Federgruen and van Ryzin [5] for a generalized bin packing problem, which is based on a partition scheme together with a set covering formulation. It can also be viewed as a modification of the classical set covering approach to vehicle routing problems due originally to Balinski and Quant [1]. The difference is that in our approach, we use a partitioning scheme to reduce the size of the set covering problem, thereby achieving efficient running times together with near optimal solution costs. By combining two powerful ideas – partitioning and set covering – from two different research areas – probabilistic analysis and math programming – we are able to provide a quite complete analysis of the solution cost of very wide class of VRPs under very general probabilistic assumptions. The approach also connects, in an appealing way, these two seemingly disparate research approaches to solving VRPs.

Overview of the Paper

In §1 we specify the properties of the class of VRPs that we analyze and give some specific examples in this class. This class is quite general, and includes problems with time window constraints, complex capacity restrictions, shift constraints, general distance metrics, inventory-routing costs, pick-up and delivery requirements and a myriad of other practical constraints and costs. Some examples in this class are formulated for illustration. In §2 we analyze a version of the problem in which the distribution of locations and attribute values is assumed to be discrete. We characterize the asymptotic optimal solution value as the value of an underlying linear program which depends on the joint probability mass function of the locations and attribute values; we also derive bounds for the tail of the minimum cost value and specify the complete limiting distribution of the minimum cost value.

In §3 we address continuous location and attribute distributions. Our approach here is to approximate the continuous distribution by a sequence of progressively finer discretization (partitions) of both the service region and the attribute space. This allows us to employ the integer programming analysis of §2. Our results are obtained by an elementary discretization based on the cubic histogram. We prove asymptotic optimality (a.s.) of an efficient (polynomial) math programming heuristic for this class of problems.

In §4 we briefl discuss computational issues surrounding our proposed heuristics. In [5], we showed that for the generalized bin packing problem, column generation techniques can often be used to solve the linear problems efficiently in practice. Further, if column generation is polynomial, one can use the Grotschel-Lovasz-Schrijver [6], [7] ellipsoid algorithm to show that the linear programs can be solved in time polynomial in the number of object types. We extend these results to a special case in our class, the classical VRP, and show that (a.s.) asymptotically optimal, polynomial time heuristic can be constructed for this problem.

References

[1] Balinski, M. and Quandt, R. (1964), "On an Integer Program for a Delivery Problem," *Operations Research*, 12, 300-304.

[2] Beardwood, J., Halton, J.H. and Hammersley, S.M. (1959), "The Shortest Path Through Many Points," *Proc. Cambridge Philos. Soc.* , 55, 299-327.

[3] Bramel, J. and Simchi-Levi, D. (1991), "A Location Based Heuristic for General Routing Problems," Columbia University Working Paper.

[4] Federgruen, A. and Simchi-Levi, D. (1992), "Analytical Analysis of Vehicle Routing and Inventory Routing Problems," To be published in *Handbooks in Operations Research and Management Science*, volume on "Networks and Distribution", M. Ball, T. Magnanti, C. Monma and G. Nemhauser, eds.

[5] Federgruen, A. and van Ryzin, G.J. (1992), "The Probabilistic Analysis of a Generalized Bin Packing Problem with Applications to Vehicle Routing and Scheduling Problems," Columbia University Working Paper, September 1992.

[6] Grotschel, M., Lovasz, L. and Schrijver, A. (1981), "The Ellipsoid Method and its Consequences in Combinatorial Optimization," *Combinatorica*, 1, 169-197.

[7] Grotschel, M., Lovasz, L. and Schrijver, A. (1988), *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin.

[8] Haimovich, M. and Rinnooy Kan, A.H.G. (1985), "Bounds and Heuristics for Capacitated Routing Problems," *Mathematics of Operations Research*, 10, 4, 527-541.

[9] Karp, R.M. (1977) "Probabilistic Analysis of Partitioning Algorithms for the Traveling Salesman Problem," *Mathematics of Operations Research*, 2, 3, 209-224.

[10] Steele, M.J. (1981), "Subadditive Euclidean Functionals and Non-Linear Growth in Geometric Probability," *Annals of Probability*, 9, 365-376.

A Parking Simulation Model for Evaluating Availability Information Service.

By

Yasuo ASAKURA

Department of Civil and Ocean Engineering

Ehime University

Matsuyama, 790, JAPAN

Parking guidance and information (PGI) systems aim to provide information to drivers concerning the location, direction and availability of parking spaces. The availability information shows the level of congestion in a car park such as FULL, CONGESTED, SPACES or CLOSED. The PGI systems are expected to redistribute parking demand among car parks and to reduce queues at the most popular car parks.

The intelligent PGI systems applying high-technology have been recently implemented in several Japanese cities. However, it is not known how tho PGI system has improved parking conditions. This paper aims to evaluate the effects of PGI systems on drivers' parking choice behavior and resulting parking performance. In particular, we focus on the differences between types of availability information, such as FULL/SPACES, expected queuing time and/or number of vacant spaces.

A simulation model is developed which consists of demand, performance and information service sub models. The demand model describes parking choice behavior of trip makers within the system. The disaggregate modeling approach which has been investigated for understanding individual travel choice behavior is applied. An improved multinominal logit model is formulated including information-related variables. Behavioral differences between informed drivers and non-informed drivers are explicitly described. A non-informed driver is assumed to have prior information on the attributes of a parking, for example, parking charge and walking distance from his/her final destination. An informed driver is assumed to be served with availability information as well as prior experience and his/her utility function consists of variables of prior information posterior availability information.

The performance sub model presents the interaction between congestion and level of service of a parking. The model is available to describe the condition in which time dependent arrival rate sometimes excesses service rate. In the information service sub model, the outputs of the performance sub model are transferred to the availability information such as FULL or SPACES. More detailed information such as queuing time and number of vacant spaces are also produced. These variables are used as the input for the demand model mentioned above.

Numerical examples are then calculated for evaluating the effects of different types of availability information service. Changing both the congestion level of parking and the ratio of informed and non-informed drivers, different availability information are given to informed drivers.

It is shown that the amount of user's time savings by PGI system depends on the ratio of numbers of informed and non-informed drivers. When the level of congestion becomes higher and almost of all car parks are occupied, FULL/SPACES information which is served in many present PGI systems would loose its merit to redistribute parking demand. The model suggests that the FULL/SPACES information should be altered by other types of information such as queuing time information when the congestion level grows higher.

A rerouteing algorithm for the Munich Outer Test Network.

by Caroline M Shield[1], Michael G H Bell[1], Hartmut Keller[2], and Thomas Sachse[2]

Abstract.

As a result of cooperation, a rerouteing strategy has been developed for the Munich Outer Test Network. It is designed to achieve the objective of a system optimum assignment which is desired for rerouteing using *variable message signs*. The algorithm iteratively reassigns trips to the set of paths with least marginal costs, updating at each interval the marginal costs that fall within a rolling horizon. Experiments using the algorithm have shown how it can adapt to sudden increases in flow by rerouteing trips across the network. Further work will include off-line testing of the strategy.

1. Objective.

Through cooperation between the Transport Operations Research Group (TORG) and the Technical University of Munich a rerouteing strategy has been developed. The strategy is designed to produce a set of proposed path flows that achieve the objective of a system optimum assignment which will then be implemented through *variable message signs*.

2. Approach.

The algorithm is iterative in nature, allowing the approximation to convergence to be monitored, an essential feature of a strategy which is seeking to attain a system optimum. The algorithm uses a time-slice approach, whereby time is divided into five minute intervals. The five minute time-slice is a feature in common with the EURO-SCOUT route guidance system from Siemens.

In addition to the time-slice approach, a rolling horizon is implemented. The horizon which encompasses three time-slices rolls forward by one time-slice. The effect of the horizon is that although each time-slice is treated in sequence, there are continually three time-slices in focus at any time. The aim is to allow for the effect of future trips on current trips.

Traffic is loaded onto the network and stored as link flow profiles in a multi-dimensional database of link flows. From the link flows and using a link specific marginal cost function, marginal costs for each link can be determined. Trips are allocated to paths with least marginal cost. This leads to a system optimum as a result of the *marginal cost pricing principle*.

3. Background.

The goal of the *rerouteing strategy* is to encourage traffic to use specific paths in the network so as to approximate a system optimum assignment. The approach used on the Munich Outer Test Network is a system

[1] Transport Operations Research Group, University of Newcastle upon Tyne, NE1 7RU UK

[2] Fachgebiet Verkehrstechnik und Verkehrsplanung, Technische Universität München, Arsisstraße 21, 80333 München, Germany

of *variable message signs (VMS)*. The network is a peri-urban network, linking the centre of Munich with commuter suburbs but it is also a main motorway link.

The BABSY/X strategy currently used by the Munich Outer Test Network has two components; the prediction of traffic flow in the network and the evaluation of the predicted flow. The network is divided into sections. The traffic passing a branching point in the network during the current 5 minute interval is moved forward through the sections along the path. The expected density for each section is determined by adding the net inflow to the base density. The speed is determined from the density. The base density is determined from speed measurements when the system is reset. When congestion is predicted to occur in any section for the current routeing, predictions are performed for all alternative routeings. These are then evaluated on the basis of the travel time and energy consumption, which is a function of speed and the vehicle type composition of the traffic of each section. The best set of routeings, subject to certain constraints, are then implemented through the variable message signs. The rerouteing strategy described here is similar to the BABSY/X approach, with the exception that density is not considered and paths are not enumerated.

4. Algorithm Description.

The rolling horizon implemented can be considered as a look-ahead period. Within this horizon the effects of the rerouteing strategy are evaluated and can be modified. Journeys entering the network or journeys commencing within the network during the current horizon will be routed in order to improve the performance of the network. Journeys extending into the horizon which started in a completed time-slice remain on their paths, since these journeys are still in progress. There is some loss of realism here, as future trips may influence trips in progress, at least in terms of their travel times. The rolling horizon is shown in Figure 1. The iterative behaviour of the strategy improves performance of the network by making continual refinements to the database of link flows.

In advance of the iterative procedure, the network details and demand data for each *origin-destination (O-D)* pair is input. The network information is stored as link data, each link has associated with it it's length, the downstream links and the speed-flow relationship type. Also loaded are the speed-flow relationships, a set of relationships for typical links in the network.

The first step once all data is loaded, is to build a set of time-dependant least marginal cost paths starting in the current time-slice, between all O-D pairs for which there is flow during the current time-slice. A form of Dijkstra's algorithm, modified to build time-dependant paths, is used to determine paths with least marginal cost.

The next stage is to load the flow onto these paths. The proportion to be loaded is calculated on the basis of the number of paths existing between the O-D pair which start in the current time-slice. Initially of course, all the O-D pair's flow is loaded onto the first set of least marginal cost paths. In each subsequent iteration, the new set of least marginal cost paths are assigned a proportion of the relevant flow.

The flow is entered into the link-flow database at the appropriate link for the appropriate start-time and current-time pair. This positioning is necessary because all the links along a path are associated with a common start-time but different current-times since the flow takes some time to propagate along the path. The travel time along a link is calculated using the average speed.

At this point the algorithm advances to the next time-slice in the horizon, and repeats the path building and flow assigning process. When this stage is completed, the third time-slice in the horizon becomes the focus. Once all three time-slices in the horizon have been completed, the second iteration is started. On completion of all the iterations of the current horizon, time is advanced and the iterative process recommences with the horizon's first time-slice moved to the second time-slice and so on. In simplistic terms, the movement of the algorithm through time can be compared to a "three steps forward and two back" movement. Minimising the total marginal costs using the iterative process optimises the system.

After the first iteration flow is split between the paths. In reality this means removing flow from the link-flow database from the positions they occupied, then replacing the new proportion of flow in the correct time-field of the database. In order to remove the flow from position in the database at which it was stored, the travel time from the last iteration must be carried with the path details. This is necessary because the travel times for the last iteration cannot be recalculated on the basis of the current database, due to the changes to the database in the interim period.

Paths which are started can still have an important influence on the network, since although started in earlier time-slices they may extend into the current time-slice. However, path flows which are completed prior to the current time-slice can be deleted. This will release memory and reduce computation time by reducing search time. These savings can be significant especially for large networks and long journey times.

Sachse et al.(1994) discuss the subject of the Fundamental Diagram with specific attention to the Munich outer test network. The paper proposes several specific speed-density relationships for different links in the network under different conditions (eg. proportion of heavy goods vehicles (HGVs) and time of day). Some of these relationships (those which exclude the influence of a high proportion of HGVs) have been adopted for use in the rerouteing strategy. A typical speed-flow curve, shown in Figure 4, is determined from the data for the link shown in Figure 3. Figure 3 suggests two possible values for speed for any one value for flow. The rerouteing algorithm considers only the higher speed, namely the line depicted by B in Figure 4. This aligns with the aim of determining the optimum assignment if higher speed can be associated with less cost (see Section 7). As a result of selecting curve B as the area of our interest, it is necessary to define how to treat area A. In area A a high marginal cost is assumed. This effectively prevents any path flows exceeding capacity.

5. Experimental Description.

An instance when the VMS system may be called into action could be a significant burst of flow between a particular O-D pair. This could have various causes, for example a diversion. To test the algorithm this situation was simulated.

The O-D pair selected is denoted by *a-c* in Figure 5. The origin *a* was chosen because it has a central position. More than one path choice exists between this pair. The routeing between this pair under normal demand is the most obvious, direct route North. However, an alternative route including link *b* would involve a significant alteration to usual flow patterns. Link *b*, in fact, is not used by the algorithm under normal circumstances.

A burst of demand was loaded between this O-D pair during the first three time-slices after which demand returned to low normal levels. The level of demand was set particularly high during the burst, but remained within region *B* shown in Figure 4. The demand profiles for O-D pair *a-c* are in Table 1.

A run of the algorithm using the new demand was executed. The results show that the increase in demand causes trips to be loaded along the previous minimum cost paths. Significantly, trips are also loaded along the alternative (and longer) alternative path, which includes link *b*. Other flows in the network are also changed, for example, the link in the opposing direction to link *b* which was not previously used, now has a small flow. Table 2 shows the flow on these two links when O-D *a-c* is subject to high demand.

The change in routeing shows how the strategy works to absorb the change in demand by routeing flow in the network according to updated marginal costs.

Table 1. Demand between O-D pair a-c.

Time-slice	0	1	2	3	4	5	6	7	8	9	10	11	12	12	14
Normal Demand	0	0	0	2	0	0	2	1	1	3	1	1	0	0	0
Increased Demand	4000	4000	4000	6	4	4	6	4	4	6	4	4	6	4	4

Table 2. Link-flows with high demand.

Time-slice	0	1	2	3	4
Flow on link b	1.5	1044.5	513.0	342.0	0.0
Flow on link in opposing direction	4.25	13.0	24.833	4.958	9.5

6. Results.

As a result of running the rerouteing algorithm a detailed output of several of the important parameters is available. The link flow database is output, showing all flows recorded in the database. This is updated at each iteration. Also recorded is the path-flow data. This shows the O-D pair, the corresponding demand, the proportion of that demand to be loaded onto the path and the path-flow itself. A file of paths which are completed is written at each iteration prior to their deletion, the remaining paths are also saved. All new paths and their initial costs are written as each set of new paths is generated. From this information, the growth of the link-flow database can be observed as paths start in new time-slices and flow is loaded onto the network. The path-flow data enables tracking of proportions and path-flows.

The output also provides an interesting history of the propagation of paths and their flows. For example, important links which are unused are unexpected particularly when they are shorter in length than an alternative routeing. When experimenting with the network by removing the alternative links (as an incident would do) or as in the experiment described above, the flow was caused to switch the routeing to the previously unused links, indicating that the network was complete and functioning correctly both under normal circumstances and extreme ones. It is clear how important the speed-flow relationships are to the network when examining the usage of particular links. An unfavourable speed-flow-relationship on one link can ensure that trips do not use a particular route at all. This was shown to be the case for some links.

One of the dilemmas when using this strategy is when to update the link-flow database. The nature of the algorithm ensures that at each stage the costs and flows are slightly out of sync; as a result of updating one the other requires an amendment. The most obvious answer is to update links and costs as often as possible. The most suitable approach to ensure that the implementation of the algorithm follows this flexible approach lies in the recalculation of travel times when required providing the most immediate method of updating, with the flows and marginal costs updated at each iteration.

7. Further improvements.

The marginal cost function is currently only a function of travel time. The introduction of a variable representing a fuel emission factor would enhance the scope of the function. However, if marginal cost does not increase monotonically with flow there could be problems with the stablility of the algorithm. Further changes to the algorithm encoding will improve the computational efficiency of the strategy. Following these amendments, an off-line comparison between the rerouteing strategy and the BABSY/X strategy is planned.

8. References.

Sachse T., Keller H. (1994); Darstellung des Verkehrsablaufs auf der BAB A9 als Basis für dir Visualisierung des Verkehrszustandes und die Validierung von Modellen zur Analyse und Prognose des Verkehrszustandes.

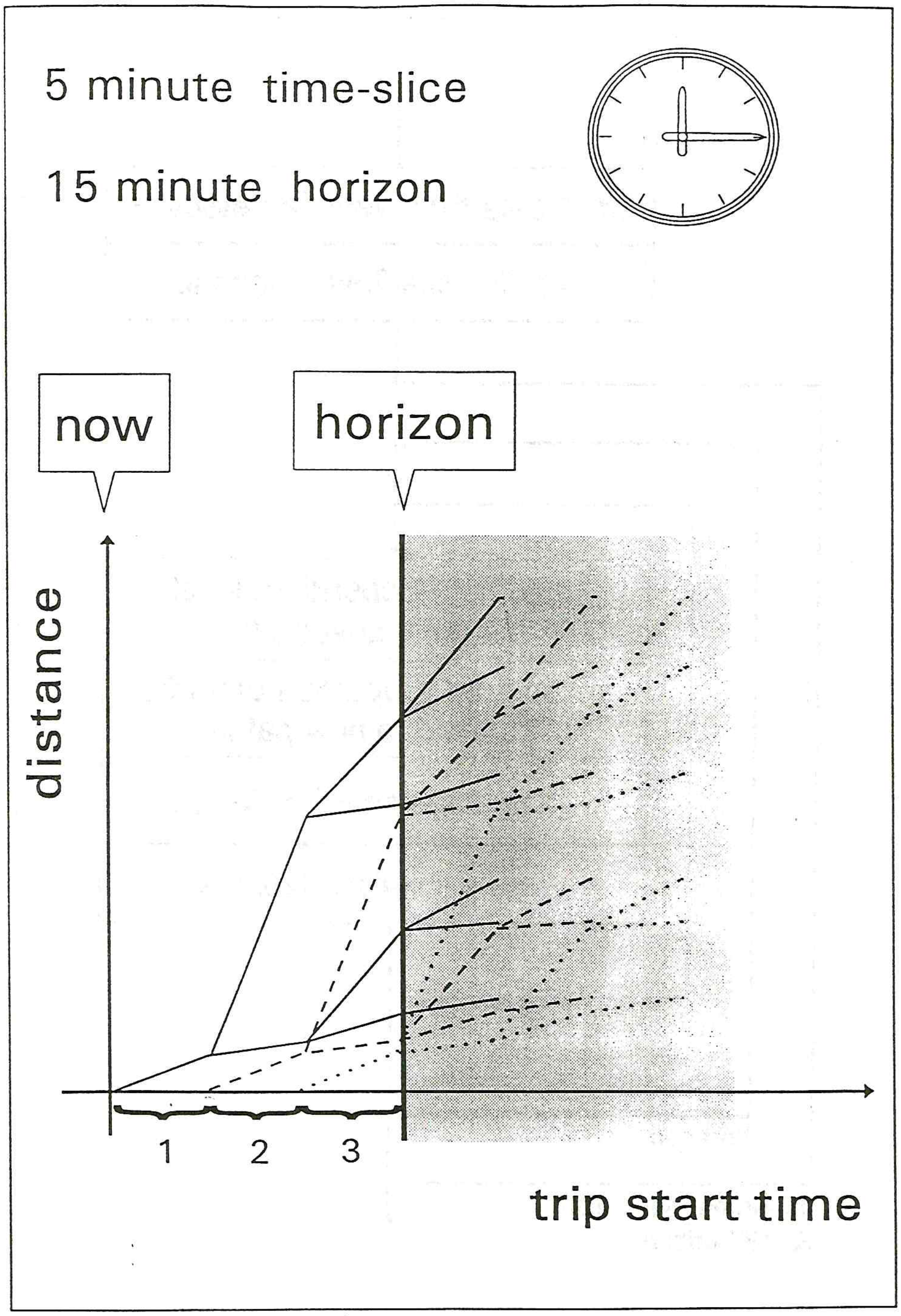


Figure 1. The rolling horizon approach.

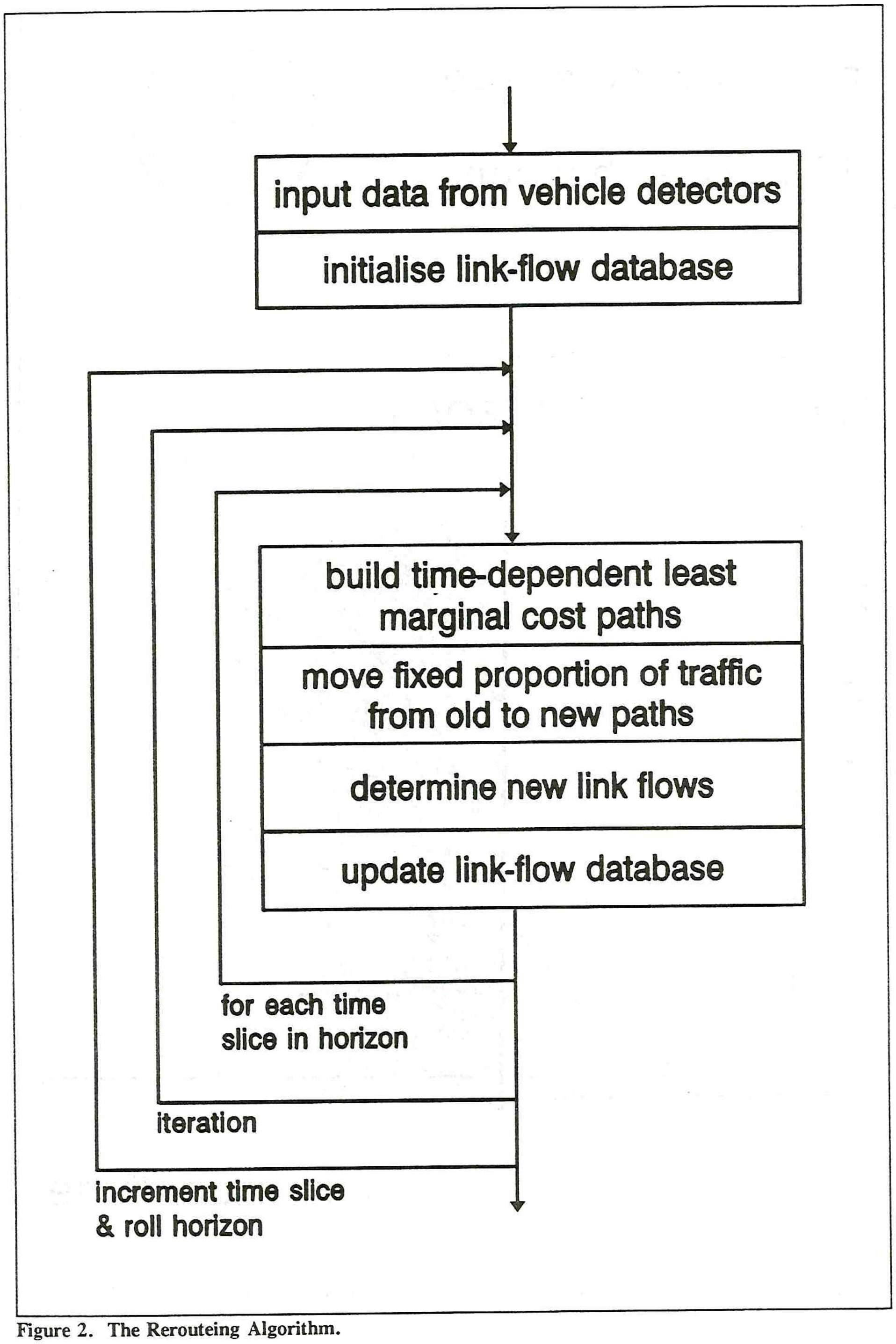

input data from vehicle detectors
initialise link-flow database
build time-dependent least marginal cost paths
move fixed proportion of traffic from old to new paths
determine new link flows
update link-flow database
for each time slice in horizon
iteration
increment time slice & roll horizon

Figure 2. The Rerouteing Algorithm.

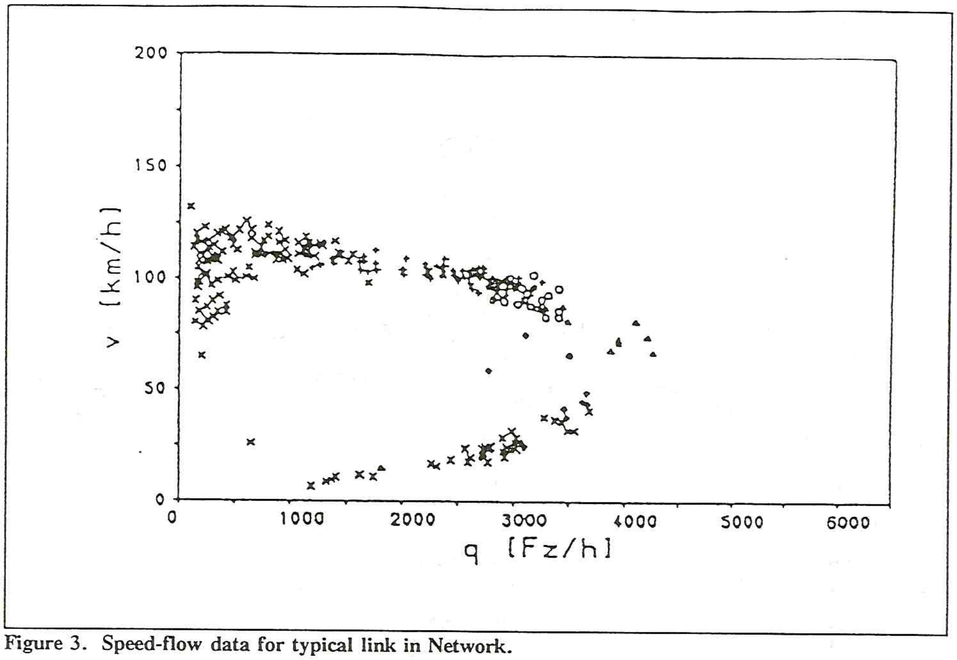


Figure 3. Speed-flow data for typical link in Network.

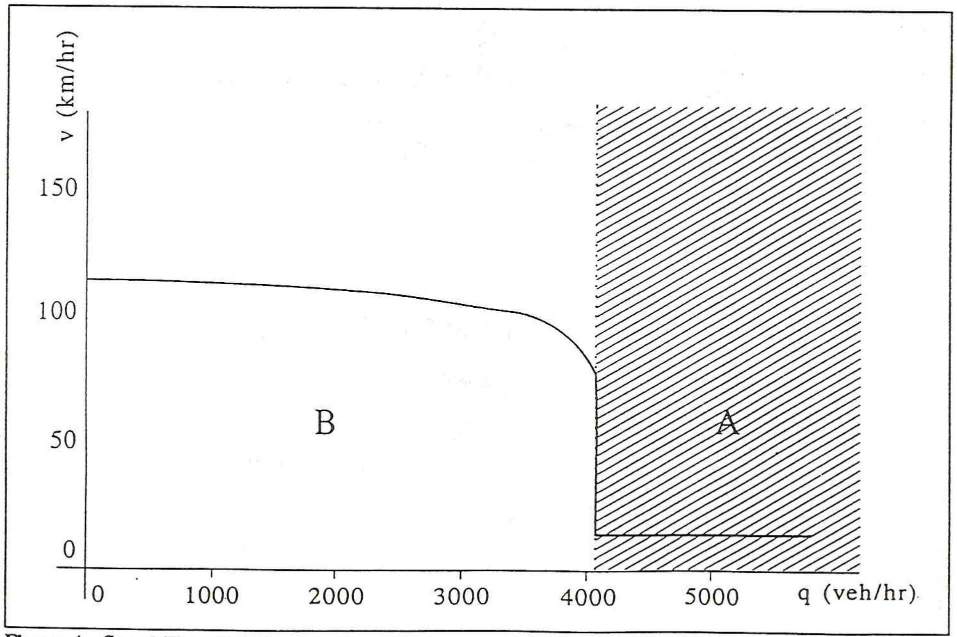


Figure 4. Speed-Flow Relationship Treatment.

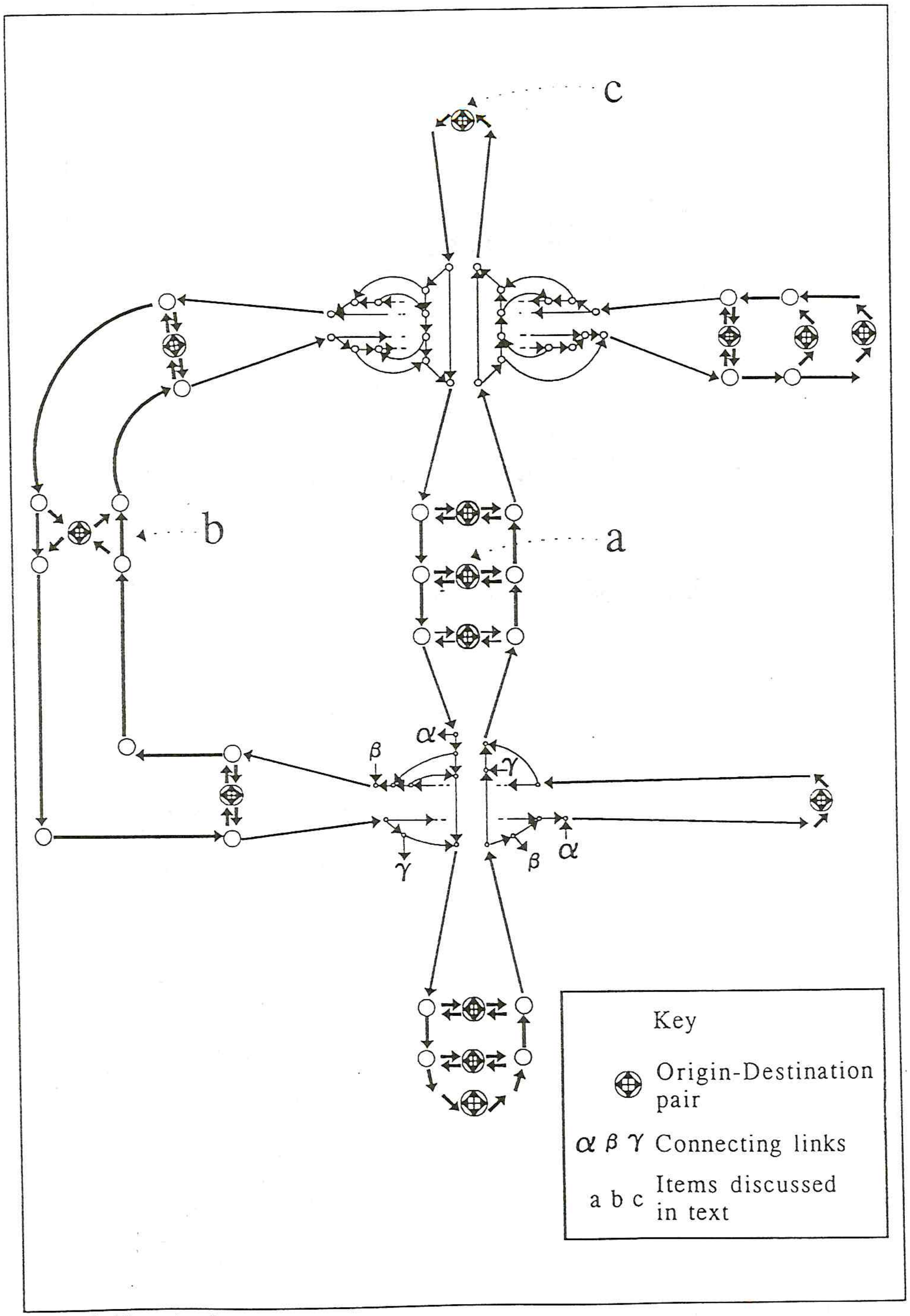


Figure 5. The network.

Dynamic Shortest Path Algorithms and Construction of Routing Tables for IVHS Applications

Haris N. Koutsopoulos, Carnegie Mellon University
Andras Farkas, MIT

The emergence of Intelligence Vehicle Highway Systems has generated renewed interest in the shortest path problem. Shortest paths are essential elements of routing strategy generation and dynamic assignment algorithms that may be used for prediction of traffic conditions. Important characteristics of the problem include the large size of the networks involved, the need to solve the all-to-many problem and the time dependent nature of link costs.

Habbal et. al. examine the potential of massively parallel architectures for the solution of the static all-to-all shortest path problem and propose a network decomposition strategy for the solution of the problem. They also discuss decomposition properties that improve the performance of the algorithm. Kaufman and Smith (1993) identified conditions under which static shortest path algorithms can be used for the solution of the dynamic shortest path problem as well. Romeijn and Smith (1991) examined existing sequential and parallel methods for solving shortest path problems and discuss possible approaches for the approximate solution of the problem using network aggregation techniques. Ziliaskopoulos and Mahmassani (1992) introduce an algorithm that solves the all-to-many dynamic shortest path problem. The algorithm, based on Bellman's principle of optimality, starts from a destination node and expands the network backwards while calculating shortest paths from all nodes to the destination node under consideration. The algorithm identifies the correct shortest paths even if the link costs do not satisfy the conditions of Kaufman and Smith.

The objective of this paper is twofold:

- Examine the performance of decomposition algorithms for the solution of the all-to-many dynamic shortest path problem and investigate the effect of the connectivity of urban networks to the performance of the algorithm.

- Develop methods for efficient construction and storage of routing tables which are used for the transmission of information to motorists, and examine the trade-offs between accuracy and shortest path calculations.

The proposed approach follows the ideas developed by Habbal et. al. but it is modified for solving the all-to-many shortest path problem. Network decomposition algorithms are appealing for several reasons: the computational advantages that may provide, especially in parallel architectures, and the fact that they are consistent with emerging IVHS system architectures. Such architectures assume a hierarchical and distributed functional organization and are based on the division of the network into subregions of uniform characteristics (Hotz et. al., 1992, Mirchadani et. al., 1992).

Decomposition of the network identifies subnetworks and associated cutset nodes (nodes at the interface of neighboring subnetworks). The original network is then modified by the introduction of virtual links: node-to-cutset, cutset-to-cutset and cutset-to-node virtual links. The algorithm proceeds in four steps:

- Find shortest paths from cutset nodes to all nodes in each subnetwork (each subnetwork consists of the original nodes and links).
- Find the shortest paths among all cutset nodes using only cutset-to-cutset virtual links.
- Find the shortest paths from all nodes to all other nodes in each subnetwork (which consists of the original links and the cutset-to-cutset virtual links).
- Find the shortest path from every node in a subnetwork to destination nodes in other subnetworks using the virtual link network. In this step the (virtual) network is acyclic and topological ordering of the nodes solves the problem efficiently.

A series of experiments (on serial architecture) were conducted to investigate the performance of the algorithm and compare it to other approaches. In the context of these experiments implementation details of the algorithm and several issues that arise due to the time-dependent nature of link costs are discussed. The performance of the algorithm is examined for both general and transportation networks and the effect of the network topology and the number of destination nodes is investigated. The numerical results indicate that for relatively small number of cutset nodes the algorithm outperforms alternative approaches. It is argued that in many cases urban transportation networks can be easily decomposed to subnetworks with few cutset nodes due to existence of natural barriers (e.g. rivers) and one-way streets. Therefore the decomposition approach has the potential to solve the problem efficiently.

Related to the solution of the shortest path problem is the construction of routing tables which are used for the transmission of shortest path information to motorists. Two approaches are presented for the construction of routing tables. The first approach (Postorder Numbering Approach, PNA) minimizes the amount of storage and information transmission and it is most appropriate for networks where traffic conditions change slowly. The Rolling Horizon Approach (RHA) overcomes most of the problems associated with PNA. The basis of RHA is that it utilizes all the information contained in shortest path trees generated by the shortest path algorithm. In particular shortest trees generated for node i at time t_o carry information about subtrees of shortest path trees corresponding to other origin nodes and other starting times.

In order to evaluate the RHA method several experiments with transportation networks were conducted. The measure of performance used, was node coverage, i.e. the number of filled entries in a routing table as a percentage of the total number of entries. For the networks examined, an average coverage between 31% and 71% was observed. The effect of several parameters of interest, such as the location of a node in the network, on coverage was examined. Finally an important aspect of the construction of routing tables, using the RHA method, is the trade-off between shortest path calculations and node coverage. Following Florian et. al. (1981) the implications of the experimental results and the above trade-off on other approaches to the solution of the dynamic shortest path problem are discussed.

References

M. Florian, S. Nguen and S. Pallotino "A Dual Simplex Algorithm for Finding All Shortest Paths," Networks 11 (1981)

M. Habbal, H.N. Koutsopoulos and S. Lerman, "A Decomposition Algorithm for Solution of the All-Pairs Shortest Path Problem on Massively Parallel Computer Architectures'" forthcoming in *Transportation Science*.

L. Head, P. Mirchadani and D. Sheppard, "A Hierarchical Framework for Real-Time Traffic Control," 71st Annual TRB Meeting (1992).

T. Hotz, H.N.Koutsopoulos, M. Ben-Akiva ... "A Distributed, Hierarchical System Architecture for ATMS and ATIS" second annual meeting of IVHS AMERICA (1992).

D. Kaufman and R. Smith "Fastest Paths in Time-Dependent Networks for IVHS Applications," *IVHS Journal* 1 (1193).

H. Romeijn and R. Smith "Notes on Parallel Algorithms and Aggregation for Solving Shortest Path Problems" IVHS Technical Report 91-03, University of Michigan (1991)

A. Ziliaskopoulos and H. Mahmassani, "A Time-Dependent Shortest Path Algorithm for Real-Time Intelligent Vehicle/Highway Systems Applications," 72nd Annual TRB Meeting (1993).

ASSESSMENT OF VEHICLE GUIDANCE SYSTEMS AND STRATEGIES BY SIMULATION

J.BARCELÓ and R.MARTÍN

Dept. d'Estadística i Investigació Operativa, Universitat Politécnica de Catalunya
Pau Gargallo 5, 08028 BARCELONA, SPAIN, Tel. +34.3.401 7033, Fax +34.3.401 7040

Abstract. This paper presents the description and first results obtained with a working prototype of a purposely built simulator to assist the design and assessment of vehicle guidance systems, mainly the ones based on cellular radio communication technology. This tool has been used to examine the benefits of dynamic route guidance systems compared to the static ones and to investigate the stability of the recommended routes with respect to changes in link travel times.

Key Words. Guidance systems; Simulation

1. INTRODUCTION

Dynamic route guidance systems are based on vehicle navigation systems that receive on time and accurate information on current and forecasted traffic conditions, which enable the in-car equipment to identify the best routes according to user requirements.

A lot of effort has been done in the past for estimating the potential benefits of such guidance systems, both, for the users of equipped vehicles, and for the whole transportation system as a function of the size of the fleet of equipped vehicles, [JEF81], [JEF87], [MAH91], [RAK89], [SOC90a], [VUR91]. An implicit underlying hypothesis in all these studies is that all the potential functionality's of the guidance system are fully operational.

A dynamic route guidance system consists mainly of a Data Collection System, an In-car Navigation System, a Communications System and a Traffic Information Centre (TIC). The Data Collection System collects in real-time the traffic data that will be used at the Traffic Information Centre for estimating the current traffic conditions prevailing on the road network, and make the predictions on the network travel times that will be broadcasted to the equipped vehicles, and used by the in-car equipment to identify the most suitable routes.

There will be two main traffic data sources: road traffic detectors and equipped vehicles. Most of the large urban areas candidate to the installation of guidance systems have already operational traffic control systems based on the values of traffic variables - flows, occupancies, speeds, queue length estimates, and so on - measured by conventional or advanced detection technologies -magnetic loops, radar, infrared, image processing, and so on -. All this available information can also be suitably processed for guidance purposes, constituent the primary set of data input to the Traffic Information Centre.

The on board equipment on each equipped car will enable it to behave as a floating car, collecting a wide variety of information on the behaviour of the vehicle as a function of traffic conditions: average speed and travel time between two well specified points along the car's route, number of stops, delays at stops, and so one. All or part of this information, as raw or processed data, will be sent from the vehicle to the Traffic Information Centre where will also be used to estimate the current traffic conditions and make the forecasts that will be broadcasted to the equipped cars, [SOC90b].

The long term objective of all vehicle guidance systems is that of relying completely on the floating car data to estimate dynamically the network state and make the traffic predictions. In that case the "quality" of the information broadcasted to the users will depend mainly on the following interrelated factors: The number of equipped cars in the network, that is, the size of the equipped fleet, the frequency of updating the network state, the horizon of the network state forecasting, that may depend on the average Origin/Destination travel time in the network, and the frequency with which the equipped cars traverse the links of the network, which determines the frequency of updating link information at the Traffic Information Centre.

This is a key aspect to be investigated for a proper design of any vehicle guidance system prior to its implementation. So far only partial simulation studies have been conducted in the scope of the ADVANCE, [BOY91 , and SOCRATES KERNEL, [SOC92], projects. The suitability of the currently available models to study the dependencies between the quality of the traffic information and the above mentioned factors was questioned by Boyce in a former paper, [BOY88]. The time horizon to develop the studies for the ADVANCE and SOCRATES KERNEL projects made not possible to undertake the research for developing the new dynamic models required for such purposes. Therefore ad hoc changes in the already existing models were made, and a specific methodology was proposed to obtain a reasonably good approximation from the proposed studies.

A common result of those studies is that the number of equipped cars required for the system to rely almost exclusively on the floating car-data is sufficiently large as to require a medium term horizon for the market to reach such a percentage of penetration (2.5 - 5% of all vehicles).

In consequence, at less a huge investment be made from the very beginning to equip a fleet of enough size, what will unlikely be the case, during a medium term horizon any dynamic route guidance system should operate using data from the two mentioned data sources: the traffic data collection system designed and installed for traffic control purposes, and therefore with features that do not fit completely the requirements of guidance, and an incomplete information supplied by the undersized fleet of equipped cars.

That means that in a transient period, that may last for several years, is very unlikely that the information for guidance be of quality enough to ensure that all the features of the system are fully operational. Therefore it is doubtful that the underlying hypothesis of the above mentioned studies on the benefits of route guidance are satisfied. That raises the question of which will actually be the benefits of route guidance until the full operation is reached.

To make inferences on the costs and benefits we need to estimate the level of service provided by the route guidance applications under different communications link data rates, and different levels of information, an estimate of the level of service can be based on the performance of interactive route guidance with respect to communication link performance and level of information, mainly in vehicle guidance systems based on cellular radio communication technology.

Another aspect that should be investigated when designing a guidance system in order to establish an optimal strategy, concerns to gain some insight on the stability of the recommended routes with respect to changes in link travel times. We want to be able to know how often and when should predictions be generated, how much better are the dynamic routes compared to routes generated by a static guidance system , etc..

Recalling the above mentioned comments of Boyce questioning the suitability of the already existing traffic models, and taking into account the results obtained by Mahmassani and Jayakrishnan, [MAH91], investigating the system performance by means of a new simulation approach, [CHA85], it becomes obvious that to get suitable answers to the former questions will require the development of new ad hoc models.

This paper presents the description and first results obtained with a working prototype of a purposely built simulator.

2. SIMULATOR DESCRIPTION

The simulator is based on two main components:

- A simplified model of Traffic Information Centre that crudely emulates the functionality of the communications component, and thus produces and sends traffic predictions, and

- A Vehicle Progress Simulator, (VPS), which "tracks" the equipped vehicles along the selected routes, and is able to emulate, if desired, the user's acceptance of the guidance information according, for instance, to Mahmassani's rules, [MAH91], or similar ones.

Both components share the information contained in a common Network Data Base, which contains, among other items, the historical link travel times.

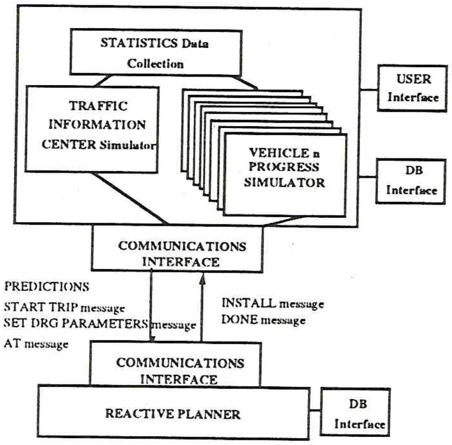


Fig. 1. The SIMULATOR modules

A communications interface is provided to interact with an external module, the Reactive Planner, which performs the planning and replanning activities, emulating, if desired, the in-vehicle process. Figure 1 displays the modular structure of the simulator.

The overall simulation will consist of the following activities :

(1) Definition of the prediction process parameters .
(2) Perform an User Equilibrium Traffic Assignment.
(3) Select (randomly) an O/D pair (s,g) for each vehicle not currently allocated to a route.
(4) Communicate s, g and the starting time t_0 to the Reactive Planner.
(5) Use Link Traversal Time functions from the library to identify shortest path from s to g starting at time t_0
(6) Record the optimal route.
(7) Simulate vehicle progression along the path.
(8) Random generation of incidents.
(9) Make new predictions.

(10) Send predictions.
(11) Identify the new route, and record it if it has changed

Activities (7) to (11) are performed several times along the trip, until the vehicle arrives at the destination node.

2.1 Network Data Base: Historical Link Traversal Times

The representation of the historical link travel times in the Data Base is one of the key questions both for our simulation purposes and for the future operation of most guidance systems. Our basic assumption is that most large cities have been measuring traffic parameters in their road networks and recording these data for years. The kind and amount of data stored varies from one city to other.

For the purposes of our system we are interested in historical link traversal times. That is, a set of functions $h_{ab_1}(t), h_{ab_2}(t), \ldots, h_{ab_n}(t)$ that for each link (a,b) give the expected traversal time for the link under certain circumstances $\{1,\ldots,n\}$. We can obtain these functions by direct measurement of the traversal times for each link or by estimating them from proper volume/delay functions when only traffic volumes for the links are available. If the volume/delay functions are very accurate, a better approximation of $tt_{ab}(t)$ can be obtained without using much dynamic information, saving on both communication bandwidth and replanning time and obtaining better initial plans.

These functions will be the result of the collection of data and its parameterization. Their parameters will be stored in the in-car equipment as well as in the Traffic Information Centre.

In our study we have estimated link traversal times from measurements of traffic volumesusing well calibrated volume delay functions for the corresponding links.

The most used volume delay functions are the Bureau of Public Road functions. These have the form:

$$u_{ab}(v_{ab}) = t_0[1 + \alpha(v_{ab} / c_{ab})^{\beta}]$$

where:

$v_{ab}(t)$ is the travel time on link (a,b) as a function of the flow volume v_{ab} on that link.
t_0 is the free flow traversal time
c_{ab} is the capacity of link (a,b)
α and β are parameters

Note that v_{ab} is normally a function of the time of day t, $v_{ab}(t)$.

Many of the large cities have these parameters calibrated for the links of their networks and flow volume measures are also commonly available.

Either traversal times or flow volume measures are taken at different intervals of time during the day. These intervals must be short enough to reflect the fluctuations of the traversal times that occur in traffic networks. Fifteen minutes seems to be a good time interval. Figure 2 is an example volume flow measure from Barcelona and the link traversal time obtained from applying the proper volume/delay function.

We assume then that 15 minutes measures of traversal times or flow volumes are available for the traffic network.

We can also obtain these values under different conditions. For instance there is normally a difference in flow volumes, and in consequence in traversal times, depending on the kind of day we make the measures, the weather conditions, etc. Measures taken on different days and under different conditions can help us to obtain the functions $h_{ab_1}(t), h_{ab_2}(t), \ldots, h_{ab_n}(t)$ that we need to approximate $tt_{ab}(t)$. (see figure 3).

2.1.2 The Parameterisation of the Functions

We need to process the data collected to find the parameters of the functions $h_{ab_1}(t), h_{ab_2}(t), \ldots, h_{ab_n}(t)$ that will be stored in the in-car equipment. In consequence, the parameterization of the functions has to:

- take into account the limited storage capacity of this equipment.
- be suitable to obtain the value of $h_{ab_i}(t)$ for any value of t.

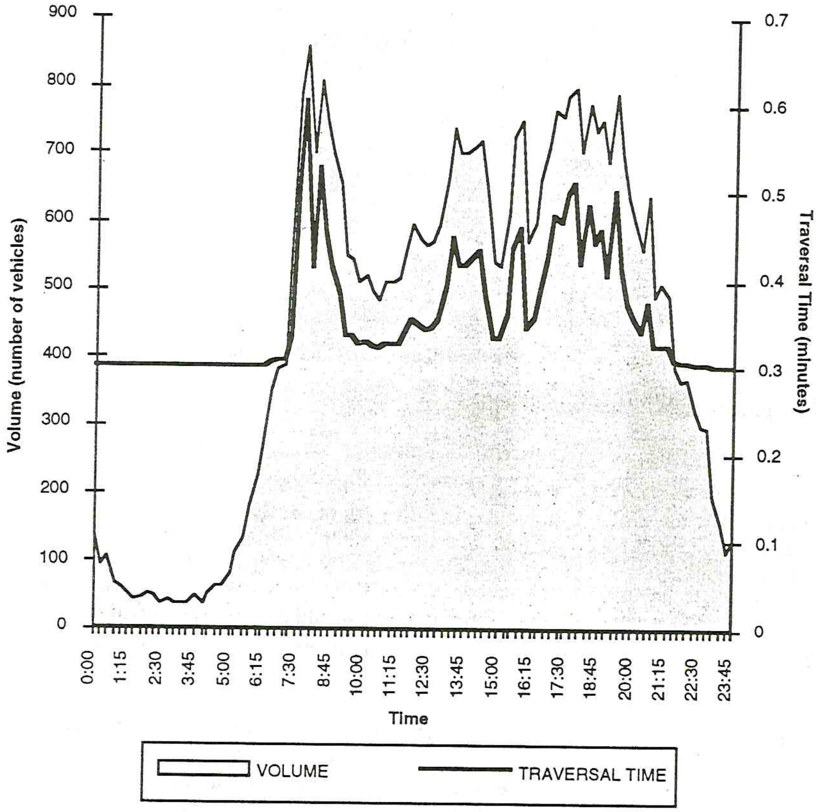


Figure 2.: Flow of vehicles measured in a link of Barcelona and traversal times obtained applying the volume delay function.

The kind of parameterization we are going to do will be a compromise between these two requirements. Due to the amount of data that will have to be processed, the parameterization method should also be as automatic as possible.

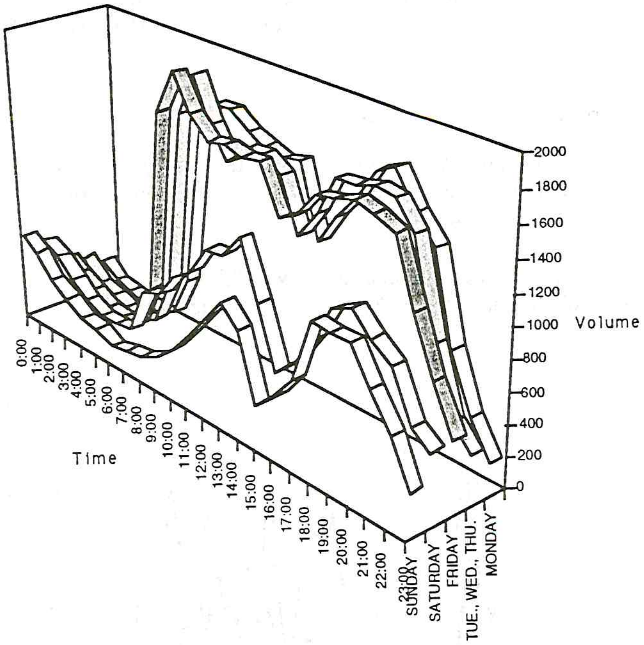


Figure 3.: Flow curves for different days of the week of a Barcelona link.

Alternative parameterization approaches

We can store the values collected every 15 minutes and interpolate them to obtain the intermediate values of the function. That means that for each function $h_{ab_i}(t)$ we will need to store 96 values. The amount of storage required for the network will be $n \times 96 \times m$, where n is the number of historical functions and m is the number of links. However, as we can see in figure 2 the traversal times are nearly constant for small volumes and in consequence we may reduce the number of values stored by recording only the values that differ significantly from t_0 (the free flow time). In the example of figure 2 we would store the traversal times between 6:15 and 22:45 together with the value of t_0, and the identifiers of the first and last interval stored.

A second approach would be to estimate a polynomial $p_{ab_i}(t)$ that fits the empirical data. With the data available from Barcelona, we have made regression analysis to find a polynomial of degree ≤ 20 that minimises the error between the estimated values and the real values. These fits always modify the shape of the function by translating the peak hours. Using them we can compress the data stored and easily obtain the values of the function but, obviously, this may cause the planner to make erronious decisions.

If we divide the range of time in intervals corresponding to the most significantly fluctuations, then it's easier to fit a curve for each of them that approximates the values and the shape of the function. However, this method is not suitable to be automatised in an easy way and implies that the function will be discontinuous in the limits of the intervals.

The proposed parameterization method

Another approach is to consider a set of p points $(t_i, h(t_i))$ and find a set of $p-1$ third degree polynomials $S_i(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, that interpolate each pair of points and that assure not only the continuity of the function but also of the first and second derivative. These polynomials are called cubic splines and we can apply well known numerical methods to find them. If we apply these methods to p points selected between the collected data we will obtain p coefficients c_i that we will use together with the values t_i and $h(t_i)$ to interpolate the function between t_i and t_{i+1}. In consequence, for each function $h_{ab_i}(t)$ we will need to store p triples $(t_i, h(t_i), c_i)$. However, if we consider equidistant values of t_i we would not need to store the t_i values.

The total amount of storage needed will be $n \times m \times p \times 3$, with n and m as before and p is the number of points selected. This can be reduced to $n \times m \times p \times 2$ in the case of equidistant values of t_i. Even though the amount of storage required is not significantly reduced, this method gives us a very good approximation of the function $h_{ab_i}(t)$.

The parameterization will then consist in:

1. select p points among the collected data
2. apply numerical methods to obtain the p c_i coefficients of the cubic splines that interpolate the values of the function between the selected points.
3. store $h(t_i)$, c_i and t_i for the p points (t_i only if required)

To obtain the value of $h_{ab_i}(t)$ we will use the information stored to:

1. determine the target interval of t: $[t_i, t_{i+1}]$, where $t_i \leq t \leq t_{i+1}$
2. calculate the value of the corresponding polynomial using t_i, t_{i+1}, $h(t_i)$, $h(t_{i+1})$, c_i and c_{i+1}.

To minimize the amount of data stored we need to reduce p, selecting the points in a smart way. An ad hoc selection procedure has been developed and tested. Results are reported in [SOC94].

The stored historical link traversal times are the basis for traffic predictions, defined in our case as a function $p_{ab}^{t_p}(t)$ that combined with a historical link traversal time function $h_{ab_i}(t)$ gives a sound approximation of the actual link traversal time function $tt_{ab}(t)$.

2.2. Definition of the prediction process parameters

- Divide T in time intervals Δt (5 minutes, 15 minutes...). This will be the frequency of network conditions updating.

- Define the prediction horizon TH
The predictions will be updated every Δt time units for a time horizon of length TH.

2.3. Simulate vehicle progression along the path

This simulation will basically consist in the generation of events when the vehicle arrives at a node. The simulator will send an AT message to the Reactive Planner and will schedule the following event.

Let's say the vehicle is at node i, time t_i and starts to travel on node (i,j). The VPS will send a message to the Reactive Planner indicating the vehicle position. That is: AT (i,t_i)

When generating next event, three kind of situations can be considered here :

a) normal conditions : The vehicle will arrive at node j in time. That means that the next VPS event will be produced at time t_j (1).

$$t_j = t_i + t_{ij}(t) \qquad (1)$$

$t_{ij}(t)$ is the travel time on link [i,j] computed from the library of functions.

b) out of route in time : The vehicle reaches node j out of time (it's a little bit faster or slower than it was expected) In this case, t_j is obtained as (2).

$$t_j = t_i + \alpha t_{ij}(t) \qquad (2)$$

where: $\alpha < 1$ when the vehicle arrives earlier than expected and $\alpha > 1$ when it arrives later

c) the vehicle is out of route : the vehicle doesn't follow link [i,j] but link [i,k] and thus the next VPS event will be produced at time t_k, following (3).

$$t_k = t_k + t_{ik}(t) \qquad (3)$$

$t_{ik}(t)$ is the travel time on link [i,k] computed from the library of functions.

2.4. Random generation of incidents (perturbations)

a) For a certain number of links (say m) repeat:

• select a link (belonging or not to the path), it can be selected from the set of k_i routes

• decide the kind of incident and its duration

- link blocking, this means infinite resistance.

- congestion building process, this will imply an abnormal increase of resistance.

This could be implemented by multiplying the resistance by a growing factor ß.

b) Identify how the incidents affect network conditions

• Perform a new user equilibrium assignment with the new resistances (using volume-delay functions)

• Identify the links whose flows have changed substantially compared to the initial assignment results (define threshold of change). Compute the new link travel times for those flows.

• Identify the set of links whose predicted travel times are significantly different from the historical ones (define travel time threshold).

2.5. Make new predictions

Generate predictions for the links affected by the incidents identified in the previous procedure. The prediction for a link will be a constant to be multiplied by the historical travel time for that link, that constant will be obtained from the travel times computed above.

The SIMULATOR has an User Interface used to input simulation parameters, to visualise partial results and to dynamically interact with the full system, and a Communications Interface allowing message passing between the VPS, the Reactive Planner and the TIC .

2.6. Diagram of States

The diagram of states of the two main modules of the simulator is displayed in figure 4. The MAIN state corresponds to the main loop of the Simulator, where the next event is selected depending on its scheduling time. Events can be of type "updating network" or "vehicle arriving at a position". The simulation time is updated to the next event time and this state is existed. The TIC module can be in three states:

- Update Network:

This state is entered when the simulation time reaches the update time. In this state a set of incidents is produced which affect the traffic network.

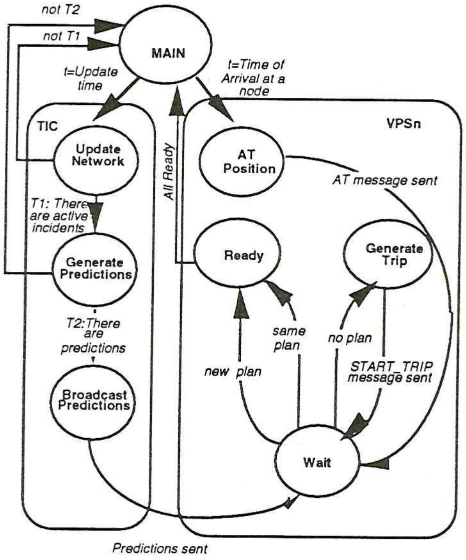


Fig. 4. The SIMULATOR diagram of states

- Generate Predictions:

This state is entered from the previous one when a set of incidents is active. Actions are performed here to identify how those incidents affect the network conditions.

- Broadcast Predictions:

This state is entered from the previous state when a set of predictions is active. A percentage of these predictions are broadcasted to all the vehicles (depending on the level of service). When the predictions have been broadcasted, all the VPS modules enter the WAIT state.
The VPS module can be in the following states:

- Ready:

This is the state corresponding to a vehicle travelling between two nodes, following a route. When all the VPS modules are in this state, the simulator enters the MAIN state.

- AT Position:

This state corresponds to a vehicle arriving at a node (in normal conditions, out of route in time or out of route). It is entered from the simulator MAIN state when the simulation time reaches the arrival time at that node. An AT message is sent to the Reactive Planner indicating the position in space and time . When this has been done, the VPS enters the WAIT state.

- Generate Trip:

This is the initial VPS state. An O/D pair is selected and a START_TRIP message is sent to the Reactive Planner. When this has been done, the VPS enters the WAIT state.

- Wait:

In this state the VPS waits for the reaction of the Reactive Planner. Two answers are possible, which are:

a) DONE, indicating to keep the current route without changes. The VPS will enter the Ready state whenever a plan is active (*same plan* condition). Otherwise (*no plan*), it will enter the Generate Trip state. That means that whenever a vehicle reaches its destination, it will be allocated a new trip.

b) INSTALL, indicating a new plan or that there is not a path between the selected O/D pair. In the first case (*new plan*), the new plan is installed and the VPS enters the Ready state. In the second case (*no plan*) it will enter the Generate Trip state.

3. PRELIMINARY ANALYSIS OF THE VARIABILITY OF THE NETWORK

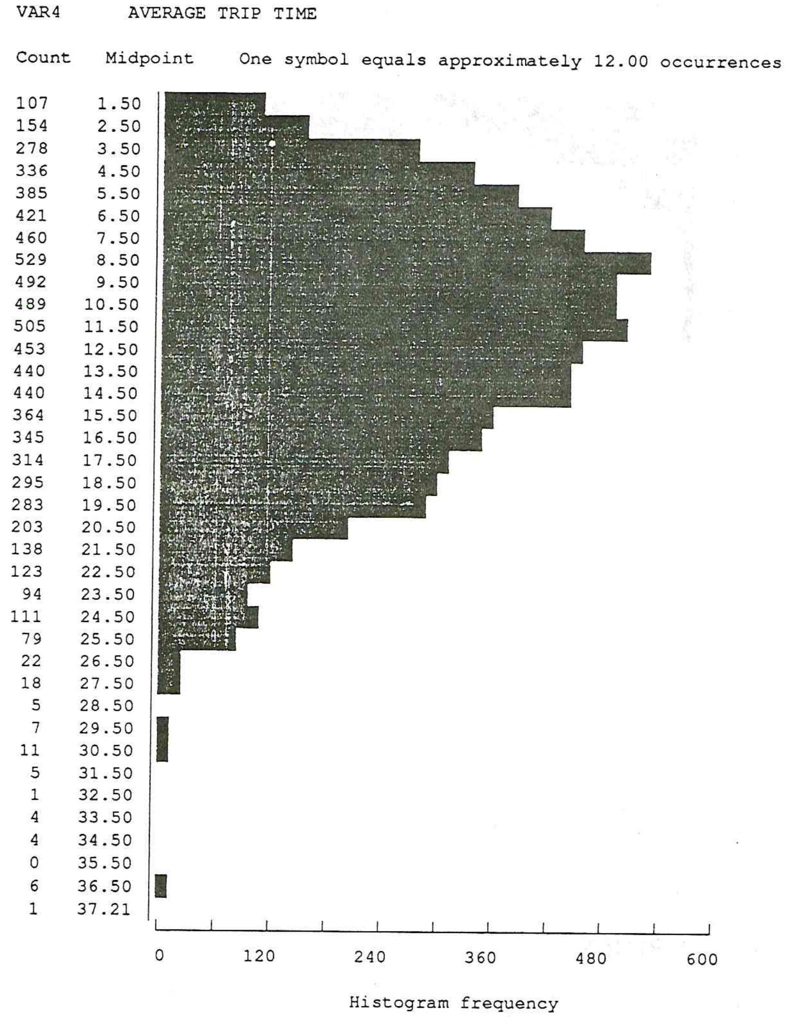
A set of simulation experiments has been conducted for the Barcelona's traffic network and a set of pre-compiled historical travel time functions for each link.

To be able to analyse the stability of the optimal routes, the optimal route in terms of time between every O/D pair of the network at each 15 minutes interval (from 0:00 until 23:45) has been obtained. The duration (trip time) of the optimal route and the number of links that form the route (hops) has also been examined. For each O/D pair the average, minimum and maximum trip time has been recorded, along with the average, minimum and maximum number of links (hops) in the route.

At each interval, the optimal route has been compared to the one obtained at the previous interval of time. The number of times that the optimal route is diffent from the previous interval one, is recorded in the variable Changes.. For each pair O/D the final value of Changes was output, giving a first idea on how stable is that route because of the historical travel times of the links which form it.

Table 1 Results of the analysis of the Network Data Base

Variables	Mean	Std Dev	Min	Max
Ave.Trip Time	12.19	5.93	1.00	37.21
Min.Trip Time	9.60	4.38	1.00	24.48
Max.Trip Time	17.70	9.63	1.00	63.13
Ave. Hops	20.76	11.45	2.00	54.90
Min. Hops	16.32	8.64	2.00	51.00
Max. Hops	28.07	15.57	2.00	66.00
Changes	24.73	64.00	0.00	13.98

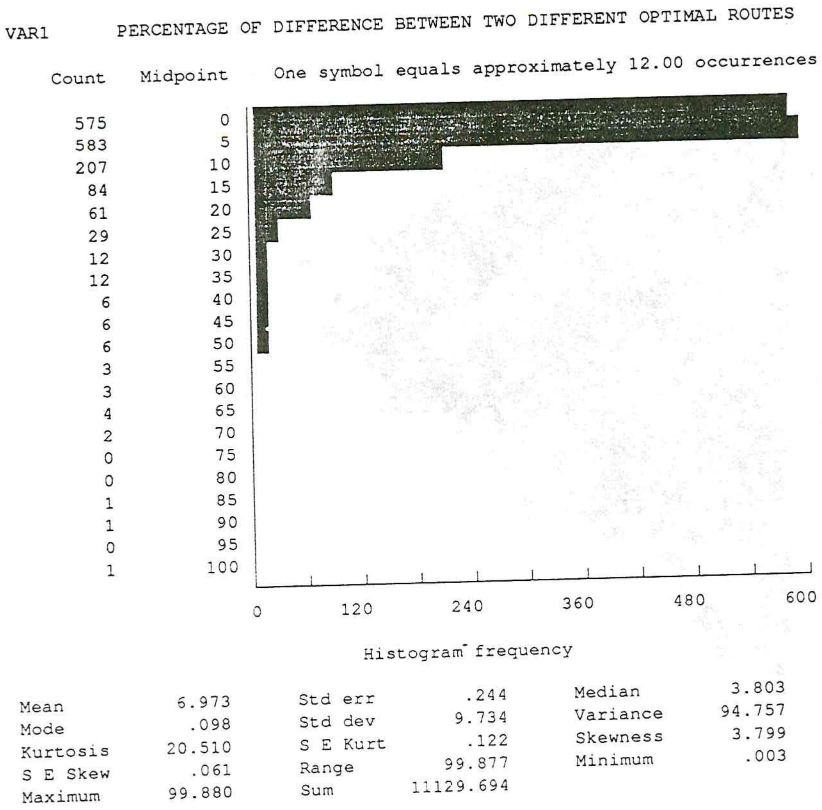


```
Mean         12.186     Std err         .067     Median       11.637
Mode          1.667     Std dev        5.926     Variance     35.122
Kurtosis      -.132     S E Kurt        .055     Skewness       .464
S E Skew       .028     Range         36.208     Minimum       1.000
Maximum      37.208     Sum        96537.425
```

Fig. 5. Histogram of the Average Trip Time

A summary of the results obtained in this way, is presented in table 1. Further information about these variables is included in figure 5.

Another variable that has been analyzed is the difference between the two different optimal routes duration obtained for two consecutive intervals. In figure 6, the results computed for a limited sample of randomly selected O/D pairs are shown.



```
VAR1      PERCENTAGE OF DIFFERENCE BETWEEN TWO DIFFERENT OPTIMAL ROUTES

   Count   Midpoint    One symbol equals approximately 12.00 occurrences

    575        0
    583        5
    207       10
     84       15
     61       20
     29       25
     12       30
     12       35
      6       40
      6       45
      6       50
      3       55
      3       60
      4       65
      2       70
      0       75
      0       80
      1       85
      1       90
      0       95
      1      100

                  0      120      240      360      480      600

                           Histogram frequency

Mean          6.973     Std err         .244     Median      3.803
Mode           .098     Std dev        9.734     Variance   94.757
Kurtosis     20.510     S E Kurt        .122     Skewness    3.799
S E Skew       .061     Range         99.877     Minimum      .003
Maximum      99.880     Sum        11129.694
```

Fig. 6. Percentage of difference between two different optimal routes.

4. SIMULATION EXPERIMENTS

A limited set of experiments have been performed using the simulator described in this paper to analyse different aspects related with Dynamic Guidance Systems.

4.1. Level of Service

This experiment was designed to measure the impact of a degraded Level of Service. In this case the Traffic Information Centre doesn't broadcast all the predictions but a percentage of them (depending on the Level of Service). The predictions to be broadcasted are selected randomly from the set of predictions. This tries to simulate situations where the equipped vehicles don't receive all the predictions because of the loose of information due to the communications system, because the Traffic Information Centre has a lack of information or because any other reason.

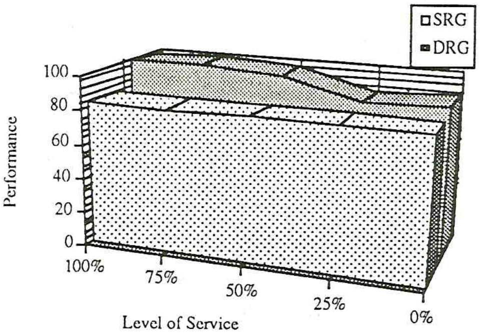


Fig. 7 Performance of the Dynamic Guidance System (DRG) and a Static Route Guidance System (SRG) related to the Level of Service

The performance in this case is measured in terms of the number of plans that the vehicles can perform during the simulation running time comparing to the number of plans done with full level of service is 100%. In figure 7 the performance obtained under different percentages of level of service is presented for the Dynamic Route Guidance system.
It is also compared to a Static Route Guidance system which uses the historical travel times to compute the optimal route but is unable to react to the dynamic information provided by the TIC or by the vehicle itself. For this later case the performance is obviously constant.

Figure 8 shows the different average trip durations obtained in the same experiment.

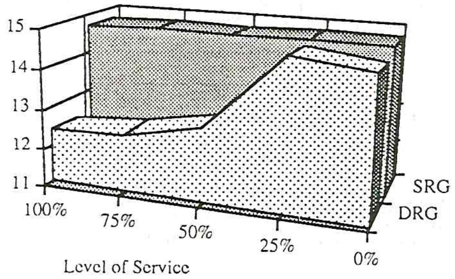


Fig. 8 Average Trip Duration obtained with a Dynamic Guidance System (DRG) and a Static Route Guidance System (SRG) related to the Level of Service

4.2. The stability of the routes

This experiment was designed to investigate how the predictions affect the stability of the recommended routes. Different simulation runs have been made here using different values for the Travel Time Threshold. The Travel Time Threshold determines when a predicted travel time is considered significantly different from the historical ones, i.e. a value of 1.10 means that the predicted travel times are at least 10% greater than the historical ones.

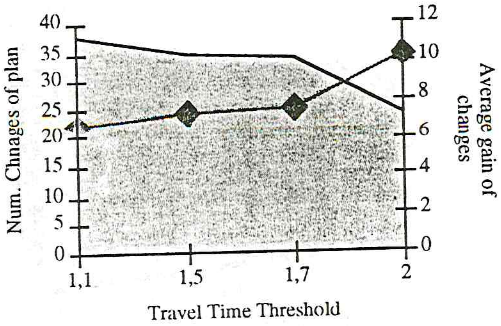


Fig.8. Number of changes of plan performed by the DRG system and average gain of changes using different Travel Time Threshold

5. CONCLUSIONS

A congested urban area like Barcelona implies a great level of variability as indicated by the preliminary analysis of the network. This can explain the benefits of using Dynamic Route Guidance systems in front of the Static ones as suggested by the preliminary results of the simulation experiments performed.

These results also indicate that the performance of the Dynamic Route Guidance system relies on the information available in the in-car equipment, which directly depends on the level of service of the guidance system. In consequence, a degraded performance of the Dynamic Route Guidance systems can be expected during a transient period until they are fully operational, in terms of the quality of the forecasts broadcasted to the vehicles.

6. REFERENCES

[BOY88] D.E. Boyce (1988). Route Guidance Systems for Improving Urban travel and Location Choices *Transpn. Res. A*, Vol. 22A, No. 4, pp 275-281.

[BOY91]. D.E. Boy, J. Hicks and A. Sen (1991). *In-vehicle Navigation Requirements for Monitoring Link Travel Times in a Dynamic Route Guidance system*, Paper presented at the 70th Annual Meeting, Transportation Research Board, Washington D.C..

[CHA85] G. Chang, H.S. Mahmassani and R. Herman (1985). Macroparticle Traffic simulation Model to Investigate Peak-Period Commuter Decision Dynamics, *Transportation research record*, 1005, pp107-121.

[JEF81] D. Jeffery (1981). *The Potential Benefits of Route Guidance*. TRRL Report, LR997.

[JEF87] D. Jeffery (1987). Route guidance and In-vehicle Information Systems. In: *Information Technology. Applications in Transport*, P. Bonsall and M. Bell (eds), VNU Science Press,
Utrecht, pp. 319-351.

[MAH91] H.Mahmassani and R. Jayakrishnan (1991). System Performance and User Response Under real-Time Information in a Congested Traffic Corridor, *Transpn. res. A*, Vol. 25A, No.5, pp. 293-307.

[RAK89] R.Rakha, M. Van Aerde, E.R. Case and A. Ugge, Evaluating the Benefits and Interactions of Route Guidance and Traffic Control Strategies Using Simulation, Proceedings of the 1st Vehicle Navigation and Information Systems Conference, Toronto, pp 296-303, 1989.

[SOC90a] SOCRATES, DRIVE I Project V1007, Commission of the European Communities, Report on WP1.2.3, 'Floating Car Data', Prepared by Hoffmann Leiter, Universitat Politécnica de Catalunya, and BASt, responsibles G.Hoffmann and J.Barceló (1990).

[SOC90b] SOCRATE, DRIVE I Project V1007, Commission of the European Communities, Report on WP1.2.3, 'Floating Car Data', Prepared by Hoffmann Leiter, Universitat Politécnica de Catalunya, and BASt, responsibles G.Hoffmann and J.Barceló, 1990.

[SOC92] SOCRATES KERNEL, DRIVE II Project V2013, Commission of the European Communities, Report SCKN/UPC 03.43,92, Prepared by Universitat Politécnica de Catalunya, responsible J.Barceló (1992).

[SOC94] SOCRATES KERNEL, DRIVE II Project V2013, Commission of the European Communities, Deliverable 7, Workpackage 7.5.3, Development of a Dynamic Route Planner, Document No. KSI4530X.130, 1994.

[VUR91] T. Van Vuren and D. Watling (1991). *A Multiple User Class assignment Model for Route Guidance*, Paper No. 910815, Presented at Transportation Research Meeting, Washington D.C.

Optimal Freight Transport Pricing and the Freight Network Equilibrium Problem

Abstract

The freight network equilibrium problem consisting of K shippers and a network of M carriers is solved for both the case when carriers act noncooperatively and when they cooperate. In the noncooperative game, the carriers select a cost-plus fixed fee tariff, where the fixed fee is charged if the shipper uses any part of the carrier's network. If carriers cooperate, then the vertically efficient coalition can offer a single tariff that can be of several forms: two-part, cost-plus, quantity-discount, or minimum-charge. Both formulations are equivalent, in the sense that an equilibrium exists for each that has identical shipper output, network flows, and shipper and carrier profits. Equilibrium network flows are obtained by solving a single level traffic assignment problem that maximizes the sum of all agents profit. The division of benefits is obtained by solving a linear programming problem. Examples are presented to illustrate the results.

Consider the freight network equilibrium problem (FNEP) consisting of multiple carriers and shippers, each acting as a noncooperative profit maximizing agent. Carriers set freight rates (tariff schedules) to maximize profit. Each shipper chooses the output level and routing that maximizes its profit, given the freight rates charged and the inverse delivered demand function for each origin-to-destination shipment.

The literature on freight networks has borrowed heavily from the urban traffic problem. Early work by Petersen and Fullerton[10] assumed fixed O-D demand and routed the traffic to minimize total transportation cost using the system equilibrium criteria. Taborga and Petersen[11] extended this to variable demands and obtained a solution that had supply-demand equilibrium with traffic routed to minimize total cost. These approaches gave the carriers no active role, nor did they address the problem of what tariffs carriers should charge and the profitability of each agent. In a world of regulated transportation this seemed like a reasonable solution.

With the emphasis on deregulation, it was necessary to consider an active role for carriers. Friesz[2], Friesz and Harker[3], and Harker[5] formulated the FNEP using game theory, where the equilibrium flows and prices are the result of each agent maximizing profit. These approaches have all resulted in models that are difficult to solve. For example, Friesz, Gottfried and Morlok[4] use a sequential shipper-carrier model, while Harker[6] uses a mixed behavior model to capture the effects of varying shipper market power. Fisk[1] minimizes the carrier(s) cost subject to constraints that describe user behavior, however, the constraints are in general nonlinear, which makes the problem difficult to solve. While the need to solve for the equilibrium flows simultaneously has been widely recognized, no computational tractable procedures were available. Furthermore, an assumption common to all the literature is that the carrier sets a linear tariff.

[1] Department of Political and Economic Science, Royal Military College of Canada, Kingston, Canada.

[2] School of Business, Queen's University, Kingston, Canada.

Our approach to the FNEP differs from that in the literature in a number of important ways. The principal difference is our assumption about the form of carrier pricing. Generally, the transportation literature has assumed that carriers use linear tariffs (that is rates independent of traffic volume). However, we allow carriers to choose tariffs which depend on volume. There are two reasons for doing so. First, the relationship between shipper and carrier is one of vertical exchange. This exchange is between two independent agents and leads to vertical relationships that are often richer and more complex than those usually studied between a firm and a group of consumers. Vertical exchange transactions are not restricted to use linear pricing as required when there are many customers, but can be contracts that may take many different forms. If carriers are allowed to choose the form of tariff, then they will never choose a constant tariff rate since, in general, constant tariff rates are suboptimal (see Hurley[7], Rey and Tirole[12], or Tirole[13]). Second, the freight transportation industry has consistently used some form of nonlinear tariff, be it two-part (tapered), quantity-discounts or quantity-forcing.

We examine two game structures which differ primarily in the way carriers behave. The first, termed the C-game, has many elements of a cooperative game among carriers, (Hurley and Petersen[8]). Carriers, individually or in coalition, set freight rates (tariff schedules), and each shipper then selects an output and a carrier or coalition of carriers, given these freight rates. The key element is that carriers are able to form coalitions. The second game structure, termed the N-game, is the full noncooperative game between shippers and carriers (Hurley and Petersen[9]). Each carrier chooses tariffs, and given these tariffs, shippers choose output and route the shipment through the freight network.

Given the more general carrier pricing, both game structures have a remarkable equivalence to a problem we term the Integrated System Optimization Problem (ISP). The ISP assumes that the shippers and carriers are controlled by a single agent, and this agent chooses a network flow vector, x^*, termed the vertically efficient flow vector, to maximize the joint profit of all carriers and shippers. Denoting the equilibrium flow vector for the C-game by x^C, and for the N-game by x^N, our basic result is that

$$x^C = x^N = x^*. \qquad (1)$$

This result is important because it allows us to calculate the FNEP flows by solving the ISP, a concave optimization problem which retains the structure of the traffic assignment problem, for which there are good algorithms.

While the ISP is important for calculating FNEP flows, it says nothing about the profit distribution among shippers and carriers. For each game structure, profit distributions are determined endogenously by solving a linear programming problem.

The solution of the ISP identifies a subset of carriers over which shipment w moves. We term these carriers the vertically efficient carriers for shipment w. In the C-game, the coalition of vertically efficient carriers offers the shipper a two-part tariffs (see Hurley and Petersen[9] for a discussion of quantity-discount, cost-plus or minimum-charge tariff structures). A two-part tariff has the general form $R(q)= b+aq$, where $R(q)$ is carrier revenue if quantity q is moved. The parameter a influences the output the shipper chooses and is set to the carrier's marginal cost for the shipment. Thus the two-part tariff has the same efficiency properties as marginal cost pricing. The parameter b effects a profit distribution between shipper and carrier and will depend primarily on whether there is competition among carriers to move the shipment, and secondly, the relative cost of these competing carriers. In general, if carrier costs are close, b will be small.

The solution of the N-game has a number of interesting properties besides the flow property in (1). Each vertically efficient carrier will offer a nonlinear cost-plus tariff of the form:

$$R_m^w(q) = \pi_m^w + c_m(q),$$

where $c_m(q)$ is the network cost incurred by carrier m to move a quantity q of shipment w, and π_m^w is the profit earned by carrier m. (In general, the two-part tariffs, quantity-discounts, and other forms of nonlinear tariffs are not optimal for the N-game formulation.) However, the profit realized by each carrier is the same in the N-game as in the C-game.

In both formulations, equilibrium tariffs must be vertically efficient and distributes the profit from each shipment so that the individual carrier's profit is maximized. The approach may be summarized as: make the pie as large as possible by using a vertically efficient tariff, and then maximize the size of each carrier's slice. Thus the original approach of assigning freight traffic to minimize total cost, or with variable demands, to maximize total system profit was correct and resulted in the proper shipper outputs and network flows.

The paper is organized as follows. First the FNEP is formulated followed by a description of the ISP and a calculation of the profitability of each shipment. Equilibrium solutions for the different forms of carrier behavior are then obtained. Finally a number of examples are presented to illustrate the theory.

1. The Freight Network Equilibrium Problem

Let K be the set of shippers and M the set of carriers. Each carrier, $m \epsilon M$, operates a network represented by the directed graph $G(N_m,A_m)$, where N_m is a set of nodes and A_m a set of links (arcs) representing carrier m's physical system. Carriers may represent different transport modes and/or competitors within a mode. Interline or transfer connections between carriers and modes are included as links in one of the carrier networks. The total transportation system is described by the directed graph $G(N,A)$, where $N = \cup N_m$ and $A = \cup A_m$. For $a \epsilon A$, let x_a be the total arc flow, and $x = (\ldots,x_a,\ldots)$, the vector of total arc flows. The total cost of shipping over arc $a \epsilon A$ is $c_a(x)$, and depends on the vector of flows on the network. We assume that $c_a(x)$ is convex in x, but do not require that it be separable in arc flows, nor do we make any assumptions about the symmetry of the Jacobean. The function $c_a(x)$ reflects the congestion on the link, and can include the effect of flows in the reverse direction or other service classes.

Shipper $k \epsilon K$ controls the set W_k of origin-to-destination (OD) shipments. Shipment $w \epsilon W_k$ moves from origin r to destination s $(r,s \epsilon N)$, with flow q_w, and has delivered inverse demand function $p_w(q_w)$. We assume that the marginal revenue is decreasing, which ensures that the revenue from each shipment is concave. Let $q^k = (\ldots,q_w,\ldots)$ be the vector of OD shipments for shipper k, and $q = (\ldots,q^k,\ldots)$ the total shipment vector. The cost of production for shipper $k \epsilon K$ is $c_o^k(q^k)$, and is assumed to be convex in the vector of shipments from shipper k. A shipper may control an individual OD shipment, or an extensive set of shipments (e.g. General Motors). The set of all OD shipments is $W = \cup W_w$.

The standard path-flow formulation is used. Associated with each OD pair w, the index $p \epsilon P_w$ will refer to a path through the network for the OD shipment w, with path flow f_p^w. In this formulation, the path-flows f_p^w are the decision variables, and we let $f_w = (\ldots,f_p^w,\ldots)$ be the vector of path flows for shipment w, and $f = (\ldots,f_w,\ldots)$ the vector of all path flows.

We define the strategy space for carrier $m \in M$ to be the set of tariff schedules or contracts R_m^w that the carrier charges for each shipment $w \in W$, with vector $R_m = (R_m^w)$. Each carrier has profit

$$\pi_m(R_m) = \sum_{w \epsilon W} R_m^w - \sum_{a \epsilon A_m} c_a(x).$$

For each shipper $k \in K$, the strategy space for each shipment $w \in W_k$ is the output q_w and the path flows f_w. The set of carriers used to move shipment w is

$$C_w = \{m | a \in A_m,\ f_p^w \delta_{a,p}^w > 0\},$$

where $\delta_{a,p}^w = 1$ if arc a is part of path p for OD pair w, and 0 otherwise. The profit for shipper k is

$$\pi_o^k(q^k,f^k) = \sum_{w \epsilon W_k} \{ q_w p_w(q_w) - \sum_{m \epsilon C_w} R_m^w \} - c_o^k(q^k).$$

Each agent has a strategy space with a payoff function defined over that strategy space. The Cartesian product of the individual strategy spaces is the strategy space of the problem. An equilibrium point is a point in the strategy space where each agent maximizes its payoff with respect to its strategy space, holding the optimal strategy choices of the other agents fixed.

2. Integrated System Problem

Before solving for an equilibrium for the FNEP, we consider the problem if the freight system is integrated or owned by a single agent. The integrated system (vertically between shipper and carriers, and horizontally amongst carriers) profit or surplus is

$$\begin{aligned}\Gamma(q,x) &= \sum_{m\epsilon M} \pi_m(R_m) + \sum_{k\epsilon K} \pi_o^k(q^k,f^k) \\ &= \sum_{k\epsilon K}\sum_{w\epsilon W_k} q_w P_w(q_w) - \sum_{k\epsilon K} c_o^k(q^k) - \sum_{a\epsilon A} c_a(x) .\end{aligned}$$

Optimal flows, which we shall call the vertically efficient flows, are found by solving the Integrated System Optimization problem (ISP):

$$\max \quad \Gamma(q,x) = \sum_{k\epsilon K}\sum_{w\epsilon W_k} q_w p_w(q_w) - \sum_{k\epsilon K} c_o^k(q^k) - \sum_{a\epsilon A} c_a(x)$$

subject to

$$q_w = \sum_{p\epsilon P_w} f_p^w$$

$$x_a = \sum_{k\epsilon K}\sum_{w\epsilon W_k}\sum_{p\epsilon P_w} f_p^w \delta_{a,p}^w$$

$$f_p^w \geq 0, \quad p\epsilon P_w, w\epsilon W_k, k\epsilon K. \qquad \text{(ISP)}$$

This concave optimization problem is in a form that is easily solved for the vertically efficient flows q^*,x^*, $\Gamma^*=\Gamma(q^*,x^*)$ and marginal cost, u_w, for shipment $w\epsilon W$, using available traffic assignment algorithms.

Clearly, there is an incentive for the carrier and shipper to move q^*, with network flows x^*, if the agents can find a mechanism to divide the additional profit. That is, if the shipper and carrier were vertically integrated, then in general total profit would be greater. In general, full economic integration of vertical firms is not necessary to achieve vertical efficiency if the two are able to contract at the vertically efficient output. The carrier's problem, then, is to find a tariff schedule that induces the shipper to choose q^*, and at the same time, allow the carrier to make as high a profit as possible. Tariffs that induce the shipper to behave as if the system were vertically integrated are said to be vertically efficient.

3. Profitability of Each Shipment

To set tariffs we need to know the joints profit generated by each shipment. While the revenue is a function of output, both production and transportation costs may be joint, and a cost allocation method is required. We shall use the arbitrary rule that prorates joint cost based on the vertically efficient flows. Other allocation methods could be used.

Following Friedman[2], we develop a notation for strategy combinations that allow the strategy for one of the players to be varied while the strategies for the remaining players is fixed. Let $(x\backslash q_w)$ denote the vector x with only changes due to the single variable q_w. Thus $c_a(x\backslash q_w)$ is the cost on arc a as a function of the single variable q_w, given the remaining variables x are fixed. Following this convention, $(x_a^*\backslash q_w)$ is the total flow on arc a as a function of q_w, given all other flows are fixed at their vertically efficient values.

Consider first the production cost. Let $c_{ow}(q_w)$ be the cost of production for shipment w. If the shipper's production costs are separable in output, then this is just the cost of producing output q_w. If the shipper's production cost are joint, let

$$c_{ow}(q_w) = c_o^k(q^{k*}\backslash q_w) - \frac{\sum_{v\epsilon W_k, v\neq w} q_v^*}{\sum_{v\epsilon W_k} q_v^*} c_o^k(q^{k*}) .$$

This allocation ensures that

$$c_o^k(q^{k*}) = \sum_{w \in W_k} c_{ow}(q_w^*).$$

To calculate the cost to transport shipment w, define

$$x_a^w(q_w) = \sum_{p \in P_w} f_p^w \delta_{a,p}^w,$$

the flow on arc a due to shipment w. If transportation costs are also prorated based on the vertically efficient flows, then the total carrier cost without shipment w is

$$c^{-w} = \sum_{a \in A, x_a^* > 0} \frac{(x_a^* - x_a^w(q_w^*))}{x_a^*} c_a(x^*),$$

while the cost of moving q_w^* is

$$c_w = \sum_{a \in A, x_a^* > 0} \frac{x_a^w(q_w^*)}{x_a^*} c_a(x^*).$$

The joint profit without shipment w is

$$\hat{\Gamma}_w = \Gamma^* - q_w^* p_w(q_w^*) + c_w + c_{ow}(q_w^*) ,$$

and the profit generated by shipment w is $\Gamma^* - \hat{\Gamma}_w$. Note that

$$\sum_{w \in W} (\Gamma^* - \hat{\Gamma}_w) = \Gamma^* ,$$

so that the profit from the individual shipments aggregate to Γ^*.

4. Noncooperative Equilibrium

The essential feature of a noncooperative game between players (shippers and carriers) is that each player has a strategy space with a payoff function defined over that strategy space. The Cartesian product of the individual strategy spaceș is the strategy space of the game. An equilibrium point is a point in the strategy space of the game where each player maximizes his payoff with respect to his strategy space, given the optimal strategy choices of the other players.

We now calculate the share of the joint profit, generated by shipment w, that each agent can claim. The vertically efficient coalition is

$$C_w^* = \{m | a \in A_m, f_p^{w*} \delta_{a,p}^w > 0\},$$

which is the set of carriers that moves the vertically efficient flow q_w^*. The shipper can use. the vertically efficient set of carriers C_w^*, or possibly some other set of carriers. A subset γ of carriers in C_w^* is feasible to excluded, if the remaining carriers $\{C_w \setminus \gamma\}$ plus possibly other carriers not in C_w^* can feasibly move the shipment. Let L_w be the set of all possible sub-sets of C_w^* that can feasibly be excluded. For each $\gamma \in L_w$, let the joint profit be $\Gamma_w^{-\gamma}$.

Let π_m^w be carrier m's profit from shipment w. If the shipper selects C_w^*, its profit from the shipment is

$$\pi_o^w = \Gamma^* - \hat{\Gamma}_w - \sum_{m \in C_w^*} \pi_m^w.$$

If she selects $\{C_w \setminus \gamma\}$, then

$$\pi_o^w = \Gamma_w^{-\gamma} - \hat{\Gamma}_w - \sum_{m \in (C_w^* \setminus \gamma)} \pi_m^w, \quad \gamma \in L_w.$$

Carrier profit, $\pi_m = 0$, $m \notin C_w^*$, since these carriers have no claim.

In the noncooperative game, each carrier in the vertically efficient set will maximize its profit π_m^w subject to the conditions that the shipper is free to select the carriers used for shipment w. Each carrier $m \in C_w^*$ simultaneously solves

$$\max \pi_m^w$$

subject to

$$\pi_o^w = \Gamma^* - \hat{\Gamma}_w - \sum_{m \in C_w} \pi_m^w$$

$$\pi_o^w \geq \Gamma_w^{-\gamma} - \hat{\Gamma}_w - \sum_{m \in (C_w \backslash \gamma)} \pi_m^w \, , \quad \gamma \in L_w \qquad (LP_w^m)$$

$$\pi_o^w, \, \pi_m^w \geq 0, \, m \in C_w$$

given the value for the other carriers π's. Note that the above constraint set defines the core of the cooperative game between shipper and carriers, however, we define it based on the noncooperative actions of the shipper. To obtain a solution to this problem, we use the following Lemma:

Lemma: Let D be a convex set in $\pi = (\pi_m)$, $m \in C$, and suppose the vector π^* solves

$$\max \Sigma_{m \in C} \pi_m$$

subject to

$$\pi \in D.$$

Then for each $m \in C$, π_m^* solves the optimization problem

$$\max \pi_m$$

subject to

$$\pi \in D, \text{ given } \pi_k^*, \, k \in C, \, k \neq m.$$

proof: Since $\Sigma_{m \in C} \pi_m$ is convex, and D a convex set in π, then

$$\Sigma_{m \in C} \pi_m^* \geq \max \pi_m + \Sigma_{k \in C, \, k \neq m} \pi_m^*$$

subject to

$$\pi_m, \pi_k^* \in D, \, k \in C, \, k \neq m.$$

Rewriting, we have

$$\pi_m^* \geq \max \pi_m$$

subject to

$$\pi \in D, \text{ given } \pi_k^*, \, k \in C, \, k \neq m,$$

which completes the proof.

Thus, in the noncooperative game between shippers and carriers, for each shipment $w \in W$, an equilibrium division of profit can be found by solving the linear programming problem

$$\pi_w^* = \max \sum_{m \in C_w^*} \pi_m^w$$

subject to

$$\pi_o^w = \Gamma^* - \hat{\Gamma}_w - \sum_{m \in C_w} \pi_m^w$$

$$\pi_o^w \geq \Gamma_w^{-\gamma} - \hat{\Gamma}_w - \sum_{m \in (C_w \backslash \gamma)} \pi_m^w \, , \quad \gamma \in L_w \qquad (LP_w)$$

$$\pi_o^w, \, \pi_m^w \geq 0, \, m \in C_w.$$

We now state the main results of the paper:

Theorem 1: Let (q^*, x^*) solve (ISP). For each shipment $w \in W$, let $\pi_w = (\pi_m^w)$ solve (LP_w), and suppose each carrier in C_w^* offers the shipper the tariff

$$R_m^w = \{ \pi_m^w + \sum_{a \in A_m, x_a^* > 0} \{ c_a(x^* \backslash q_w) - \frac{(x_a^* - x_a^w(0))}{x_a^*} c_a(x^*) \} \} \delta_m^w,$$

to move shipment w, where $\delta_m^w = 1$ if $x_a(q_w) > 0$, $a \in A_m$, and 0 otherwise. Then $\{R_m, m \in M$ and $(q^*, \Gamma)\}$ is a noncooperative equilibrium for the K-shipper, M-carrier problem, that is vertically efficient.

Note that the tariff offered by each carriers is a cost-plus tariff, where π_m^w is a fixed fee equal to the profit carrier m receives if the shipment w uses any part of carrier m's network. The terms in the summation is the cost carrier m incurs moving q_w, if the costs $c_a(x^*)$ are prorated on a flow basis.

proof: Shipper k's profit from shipment w is

$$\pi^{k}_{ow}(q_w,f_w) = q_w p_w(q_w) - \sum_{m\epsilon C_w} R^{w}_{m} - c_{ow}(q_w)$$

$$= q_w p_w(q_w) - \{\sum_{m\epsilon C_w} \pi^{w}_{m} + \sum_{a\epsilon A, x^{*}_{a}>0} \{c_a(x^{*}\backslash q_w) - \frac{(x^{*}_{a} - x^{w}_{A}(0))}{x^{*}_{a}} c_a(x^{*})\}\}\delta^{w}_{m}$$

$$-\{c^{k}_{o}(q^{k*}\backslash q_w) - c_{ow}(q_w)\}.$$

If the shipper uses the vertically efficient carriers, then

$$\pi^{k}_{ow}(q_w,f_w) = \Gamma(q^{*},x^{*}\backslash q_w,f_w) + constant,$$

with maximum at q_w^{*},f_w^{*}, the vertically efficient output and path flows. Suppose the shipper selects another set of carriers $\{C_w\backslash\gamma\}$, $\gamma\epsilon L_w$, then the change in shipper profit is

$$\Gamma^{-\gamma}_{w} - \sum_{m\epsilon\{C_w\backslash\gamma\}} \pi_m - \Gamma^{*} + \sum_{m\epsilon C_w} \pi_m = \sum_{m\epsilon\gamma} \pi_m - (\Gamma^{*} - \Gamma^{-\gamma}_{w}) \leq 0.$$

Therefore there is no incentive for the shipper to switch from using the vertically efficient carriers C_w^{*} to move shipment w, and the set of tariffs is vertically efficient.

Now consider the carriers. Suppose the shippers send the vertically efficient shipments q^{*}, routed according to f^{*}. Shipment $w\epsilon W$ is moved by carriers C_w^{*}, defined by the vertically efficient path flows f_w^{*}. If each carrier receives profit from shipment w that satisfy LP_w, then by the Lemma, no carrier is able to increase its profit from that shipment, and the proof is complete.

5. Cooperative Equilibrium

In the cooperative formulation, carriers are able to form coalitions. Carriers, individually or in coalition, set a tariff for each shipment, and shippers then select output and the carrier or coalition of carriers used, given these freight rates. Hurley and Petersen[8,9] show the following:

Theorem 2: Let (x^{*},u^{*}) solve (ISP). For each shipment $w\epsilon W$, let π_w^{*} solve (LP_w). Suppose the coalition of carriers C_w^{*} offers the shipper a tariff of one of the following forms:

cost-plus

$$R_w(q_w) = \pi_w^{*} + c(x^{*}\backslash q_w) - c^{-w}.$$

two-part

$$R_w(q_w) = b_w + a_w q_w$$

where

$$a_w = u_w,$$
$$b_w = \pi_w^{*} + \mu_w(q_w^{*}),$$

and

$$\mu_w(q_w) = c_w - u_w q_w.$$

quantity-discount

$$R_w(q_w) = p_\ell q_w, \quad q_w < q_w^{*}$$
$$= p_h q_w, \quad q_w \geq q_w^{*}$$

where

$$p_h = \{\pi_w^{*} + c_w\}/q_w^{*}$$

and $p_\ell > p_h$ is sufficiently large such that if $q_w < q_w^{*}$, the shipper's profit from shipment w is less than π_o^{w}.

minimum-charge

$$R_w(q_w) = R_o \quad , \quad q_w \leq q_w^{*}$$
$$= R_o + u_w(q_w - q_w^{*}) \quad , \quad q_w \geq q_w^{*}$$

where

$$R_o = \pi_w^{*} + c_w.$$

Then $\{R_w$, $w\epsilon W$ and (q^{*},x^{*}) is an equilibrium for the K-shipper, cooperative M-carrier problem, that is vertically efficient.

The cooperative formulation has appeal in that it shows that cooperation amongst carriers does not lead to any inefficiencies the transportation system. The noncooperative formulation is appealing in that it shows that no arrangement between carriers is required to achieve a vertically efficient equilibrium, and price supports are obtained.

Both formulations are equivalent, in the sense that an equilibrium exists for each that has identical shipper outputs and network flows. This result can be generalized. We have structured the problem so that the carrier's strategy is to set tariffs, while the shipper's strategy is to select output and the routing for each shipment. This give the carriers the "lead" role with the shipper "shopping" for the lowest cost alternative. This formulation seems to best describe practice in the freight industry, however, we note that there may be exceptions, such as the auto industry, where the shipper is the leader and can force carriers to accept minimal profits by maximizing π_o^w subject to the constraints in LP_w. We can even go one step further in generality. The constraints in LP_w define the core of the cooperative game between shipper and carriers, which von Neumann and Morgenstern argue to be reasonable solutions for such games. The integrated system is the grand coalition. A necessary condition for a division of profit to be in the core is that shipper outputs and network flows be vertically efficient. The power of vertically efficient tariffs is that group rationality can be attained by individual agents acting in their best own interest.

6. Examples

In this section we present simple examples to illustrate the theory. We begin with the case where two carriers with similar costs compete to move a shippers product to market.

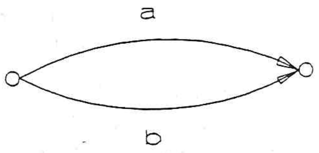


Figure 1: Competing Carriers

Example 1: One Shipper-Two Carriers, Traffic on Both Carriers

Consider two carriers competing to move a shipper's freight as shown is Figure 1. Let carrier a be the lower cost carrier with cost function $c_a(x_a)=4x_a+0.05x_a^2$, and carrier b, the higher cost carrier, has cost $c_b(x_b)=5x_b+0.05x_b^2$. Let $p_o(q)=20 - 0.2q$ and $c_o(q)=5q$. Carrier a is the low cost carrier, but costs are close enough that the vertically efficient flow is split over both carriers with $x_a^*=16.67$, $x_b^*=6.67$ and $\Gamma^*=125$. If carrier a was not available, then the traffic would move over carrier b, with $x_b^{-a}=20$ and $\Gamma^{-a}=100$. Similarly, if carrier b were not available then all the traffic would move over carrier a, with $x_a^{-b}=22$ and $\Gamma^{-b}=121$. The share of the surplus each player can claim is obtained by solving

$$\pi^* = \max \pi_a + \pi_b$$

s.t.

$$\begin{aligned} \pi_o + \pi_a + \pi_b &= \Gamma^* = 125 \\ \pi_o + \pi_a \quad\quad &\geq \Gamma^{-b} = 121 \\ \pi_o \quad\quad + \pi_b &\geq \Gamma^{-a} = 100 \\ \pi_o, \pi_a, \pi_b &\geq 0 \end{aligned}$$

with solution $\pi_o = 96$, $\pi_a = 25$, and $\pi_b = 4$. The above constraint set defines the core of the 3-person game which is shown in Figure 2. The following tariffs are vertically efficient:

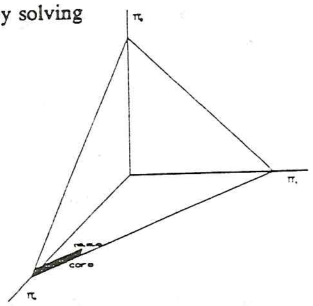


Figure 2: The Core of the Game

Individual carrier pricing:

Each carrier offers the following tariff schedules:

$$R_a(x_a) = 25 + 4x_a + 0.05x_a^2$$
$$R_b(x_b) = 4 + 5x_b + 0.05x_b^2.$$

Coalition pricing:

The coalition {a,b} offers the shipper one of the following tariffs:

cost-plus

$$R(q) = 29 + 4q + 0.05q^2 \qquad q \leq 10$$
$$= 126.5 - 5.5q + 0.025q^2 \qquad q \geq 10$$

two-part

$$R(q) = 12.89 + 5.67q$$

quantity-discount

$$\begin{array}{llr} R(q) & = 6.3\, q & q < 23.33 \\ & = 6.219\, q & q \geq 23.33 \end{array}$$

minimum-charge

$$\begin{array}{llr} R(q) & = 145.11 & q < 23.33 \\ & = 145.11 + 5.67(q\text{-}23.33) & q \geq 23.33. \end{array}$$

For each tariff, $q^{*}=23.33$, $x_a=16.67$, $x_b=6.67$, $\pi_o=96$, $\pi_a=25$, and $\pi_b=4$.

Example 2: Two Shippers-Two Carriers

Consider the problem in Example 1 with two shippers. The data is the same as in that example, except that shipper 2 has inverse demand function $p_2(q_2)=20 - 0.3q_2$ and production cost $c_o^2(q_2)=6q_2$. Solving (ISP) we obtain: $q_1^{*}=21.90$, $f_a^1=11.00$, $f_b^1=10.90$, $f_a^2=11.41$, $q_2^{*}=12.93$, $f_b^2=1.52$, $x_a=22.41$, $x_b=12.42$ and $\Gamma^{*}=178.88$. The joint profit without shipment 1 is $\hat{\Gamma}_1=63.90$ and without shipment 2 is $\hat{\Gamma}_2=114.98$.

For shipment 1, if carrier a were not available, then $\Gamma_1^{-a}=167.99$, while if b were not available then $\Gamma_1^{-b}=168.19$. The profit distribution from shipment 1 is obtained by solving

$$\pi_1^{*} = \max \pi_a^1 + \pi_b^1$$

s.t.

$$\begin{array}{l} \pi_o^1 + \pi_a^1 + \pi_b^1 = \Gamma^* - \hat{\Gamma}1 = 114.98 \\ \pi_o^1 + \pi_a^1 \quad \geq \Gamma_1^{-b} - \hat{\Gamma}_1 = 104.29 \\ \pi_o^1 \quad + \pi_b^1 \geq \Gamma_1^{-a} - \hat{\Gamma}_1 = 104.10 \\ \pi_o^1,\ \pi_a^1,\ \pi_b^1 \geq 0 \end{array}$$

with solution $\pi_o^1 = 93.41$, $\pi_a^1 = 10.88$, and $\pi_b^1 = 10.69$.

For shipment 2, if carrier a were not available, then $\Gamma_2^{-a}=167.99$, while if b were not available then $\Gamma_2^{-b}=168.19$. The profit distribution from shipment 2 is obtained by solving

$$\pi^{*} = \max \pi_a^1 + \pi_b^2$$

s.t.

$$\begin{array}{l} \pi_o^2 + \pi_a^2 + \pi_b^2 = \Gamma^{*} - \hat{\Gamma}_2 = 63.90 \\ \pi_o^2 + \pi_a^2 \quad \geq \Gamma_2^{-b} - \hat{\Gamma}_2 = 63.68 \\ \pi_o^2 \quad + \pi_b 2^1 \geq \Gamma_2^{-a} - \hat{\Gamma}_2 = 51.80 \\ \pi_o^2,\ \pi_a^2,\ \pi_b^2 \geq 0 \end{array}$$

with solution $\pi_o^2 = 51.58$, $\pi_a^2 = 12.10$, and $\pi_b^2 = 0.22$.

It should be noted that there are multiple equilibrium. In this example, the solution to ISP, (q,x), is unique, however, the flow allocation f is not. Each flow allocation results in a different equilibrium.

Example 3: Interlining

Consider the network shown in Figure 3, where one path involves interlining the shipment over arcs a and b. Suppose the demand and shipper cost function data is the same as example 1, with arc cost $c_a(x_a)=4x_a+.05x_a^2$, $c_b(x_b)=5x_b$, $c_c(x_c)=9x_c+.06x_c^2$, and $c_d(x_d)=12x_d$. The integrated system optimization problem has solution $q^{*}=13.2$, $x_a^{*}=x_b^{*}=7.2$, $x_c^{*}=6.0$, $x_d^{*}=0$, and $\Gamma^{*}=39.6$.

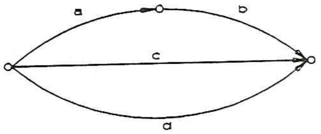


Figure 3: Interlined Carriers with Competition

The equilibrium pricing will depend on the carrier ownership of the transportation arcs. We consider two alternatives:

A. Different Ownership of Each Arc

Assume each arc is a different carrier. Then $C_w^{*}=\{a,b,c\}$. The set L_w is the sub-set of C_w^{*} that can leave the coalition and the remaining carriers can feasibly move the shipment. In this problem $L_w = \{a,b,c,ab,ac,bc,abc\}$, with $\Gamma^{\sim a} = \Gamma^{\sim b} = \Gamma^{\sim ab} = 34.615$, $\Gamma^{\sim c}=36$, $\Gamma^{\sim ac} = \Gamma^{\sim bc} = \Gamma^{\sim abc} = 11.25$. The sharing of the surplus is determined by solving

$$\pi^{*} = \max \pi_a + \pi_b + \pi_c$$

s.t.

$$\begin{array}{llll}
\pi_o + \pi_a + \pi_b + \pi_c & = \Gamma^{*} & & = 39.6 \\
\pi_o \quad\;\; + \pi_b + \pi_c & \geq \Gamma^{\sim a} & = 34.615 & \\
\pi_o + \pi_a \quad\;\; + \pi_c & \geq \Gamma^{\sim b} & = 34.615 & \\
\pi_o \qquad\qquad + \pi_c & \geq \Gamma^{\sim ab} & = 34.615 & \\
\pi_o + \pi_a + \pi_b & \geq \Gamma^{\sim c} & = 36 & \\
\pi_o \quad\;\; + \pi_b \qquad \geq \Gamma^{\sim ac} & = 11.25 & & \\
\pi_o + \pi_a & \geq \Gamma^{\sim bc} & = 11.25 & \\
\pi_o & \geq \Gamma^{\sim abc} & & = 11.25 \\
\pi_o, \pi_a, \pi_b, \pi_c \geq 0 & & &
\end{array}$$

with a solution $\pi_o=31.015$ and multiple solutions $\pi_a=4.985\alpha$ and $\pi_b=4.985(1-\alpha)$, $0\leq\alpha\leq 1$, and $\pi_c=3.6$. Carrier d is not in the vertically efficient coalition and has no claim to any of the joint surplus. The division of profit between carrier a and carrier b is not determined uniquely. Any division in the above range results is an equilibrium. In practice the division of benefits between interlining carriers is usually governed by industry rules.

B. Common Ownership of Arcs a and c

Now suppose that carrier 1 owns both transportation arc a and c, carrier 2 owns arc b, and carrier 3 owns arc d. Now $C_w^{*}=\{1,2\}$ and $L_w = \{1,2,1\text{-}2\}$, with $\Gamma^{\sim 1} = \Gamma^{\sim 1\text{-}2} = 11.25$, and $\Gamma^{\sim 2} = 34.615$. The sharing of the surplus is obtained by solving the LP

$$\pi^{*} = \max \pi_1 + \pi_2$$

s.t.

$$\begin{array}{lll}
\pi_o + \pi_1 + \pi_2 = \bar{\pi}_w\Gamma^{*} = 39.6 & & \\
\pi_o \quad\;\; + \pi_2 \geq \Gamma^{\sim 1} & & = 11.25 \\
\pi_o + \pi_1 \qquad \geq \Gamma^{\sim 2} & & = 34.615 \\
\pi_o \qquad\qquad\;\, \geq \Gamma^{\sim 1\text{-}2} & & = 11.25 \\
\pi_o, \pi_1, \pi_2 \geq 0 & &
\end{array}$$

with a solution $\pi_o=11.25$ and multiple solutions $\pi_1=23.365+4.985\alpha$ and $\pi_2=4.985(1-\alpha)$, $0\leq\alpha\leq 1$. Note that carrier 1 has much stronger market power, and the shipper's profit drops from 31.015 to 11.25.

This example clearly demonstrates that the level of freight rates for a particular shipper will depend on the ownership structure of the network. If a shipper has at least two carrier routing options which are independently owned, and the costs of these alternatives are relatively close, then the shipper's rates will be low. On the other hand, rates will be high if there is only one option or the second best route has a high cost relative to the low cost route. This example also illustrates that the efficiency of the transportation system is not affected by the pattern of carrier ownership, since a vertically efficient tariffs is always used.

Discussion

Equilibrium flows for the FNEP are obtained by solving ISP, a concave traffic assignment problem. For each shipment, the division of profit in the cooperative formulation is obtained by solving a linear program. This division of profit is also a solution for the noncooperative formulation. Equilibrium tariffs for the noncooperative game must be of the cost-plus form, while if carriers cooperate then two-part, quantity-discount and minimum-charge tariffs may also be used.

One point that should be emphasised is that there may be multiple equilibria. In solving ISP, the shipper outputs and arc flows may be unique, however, the path flows are not, a well known property of the traffic assignment problem. Each different set of path flows can result in a different division of benefits amongst the agents. However, all equilibria obtained will be vertically efficient.

To calculate tariffs, each carrier must know the shipper demand function and the competing carrier cost functions. The total flow on each arc together with the flow due to each shipment can be observed. With this data, the carrier can calculate the value of the shipment and the share each agent can claim. Thus each carrier can individually post a tariff of the cost-plus form which results in a vertically efficient equilibrium.

As the examples show, carrier profit depend on the closeness of the competitor's cost functions. Depending on the pattern of ownership of the transportation system, profit can vary substantially. However, ownership has no impact on the efficiency of the freight transportation system, which implies that the strategic role for carriers is limited.

References

1. Caroline S. Fisk, "A Conceptual Framework for Optimal Transportation Systems Planning with Integrated Supply and Demand Models", Trans. Science, Vol. 20, No. 1, pp.37-47, 1986.

2. Terry L. Friesz, "Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities', Trans. Research A, Vol. 19A, No. 5/6, pp.413-427, 1985.

3. Terry L. Friesz and Patrick T. Harker, "Freight Network Equilibrium: a Review of the State of the Art", Chapter 7, Analytical Studies in Transport Economics, Andrew Daugherty editor, Cambridge University Press, 1985.

4. Terry L. Friesz, Joel A. Gottfried and Edward K. Morlok, "A Sequential Shipper-Carrier Network Model for Predicting Freight Flows", Trans. Science, Vol. 20, No. 2, pp. 80-91, 1986.

5. P. T. Harker, Predicting Intercity Freight Flows, VNU Science Press, Utrecht, The Netherlands, 1987

6. Patrick T. Harker, "Multiple Equilibrium Behaviors on Networks", Trans. Science, Vol. 22, No. 1, pp.39-46, 1988.

7. W. J. Hurley, "A Theory of Rail Freight Pricing", unpublished Ph.D. dissertation, School of Business, Queen's University, Kingston, (1988).

8. W. J. Hurley and E. R. Petersen, "Nonlinear Tariffs and Freight Network Equilibrium", Transportation Science, forthcoming.

9. W. J. Hurley and E. R. Petersen, "Freight Network Equilibrium and Vertically Efficient Tariffs", School of Business Working Paper No. 93-17, Queen's University at Kingston, 1993.

10. E. R. Petersen and H. V. Fullerton, The Railcar Network Model, Canadian Institute of Guided Ground Transport, No. 75-11, Queen's University, Kingston, June 1975, pp. 1x + 306.

11. P. Taborga and E. R. Petersen, RAIL User's Manual, The World Bank, Washington, D.C., October 1982, pp. 105.

12. P. Rey and J. Tirole, "The Logic of Vertical Restraint", American Economic Review, 76, 921-39 (1986).

13. Jean Tirole, A Theory of Industrial Organization, MIT Press (1989).

Why Regulate Prices in Freight Transportation Markets?

W.J. Hurley [*]
The Royal Military College of Canada

E.R. Petersen [†]
Queen's University

May 1994

Abstract

We examine the efficiency of Marginal Cost Pricing and Ramsey Pricing relative to Deregulated Prices, those which obtain when carriers and shippers (producers) are free to contract without regulatory impediment. Our model suggests that:

- Marginal Cost Prices will not enhance economic efficiency relative to Deregulated Prices; and
- Ramsey Prices will degrade economic efficiency relative to Deregulated Prices.

1 Introduction

In the future, it is unlikely that North American transportation firms, particularly railroads, will escape the well-meaning scrutiny of economists and

[*] Department of Political and Economic Science, The Royal Military College of Canada, Kingston, Ontario, Canada, K7K 5L0. Phone: 613-541-6468, Fax: 613-541-6315, E-mail: hurley_w@rmc

[†] School of Business, Queen's University, Kingston, Ontario, Canada. Phone: 613-545-2349, Fax: 613-545-2321

legislators bent on ensuring that these firms do not take advantage of market power, especially in markets where producers are captive. To suggest otherwise ignores the lessons of economic history and the ebb and flow of regulatory sentiment.

In this paper we examine the efficiency of Marginal Cost Pricing (MCP) and Ramsey Pricing (RP) relative to Deregulated Prices (DP), those which obtain when carriers and shippers (producers) are free to contract without regulatory impediment. Our model suggests that:

- Marginal Cost Prices will not enhance economic efficiency relative to Deregulated Prices; and

- Ramsey Prices will degrade economic efficiency relative to Deregulated Prices.

These are strong statements which fly in the face of the conventional wisdom about regulation in freight transportation markets.

The key to these results is the way we model shipper-carrier interaction. Most, if not all, welfare analyses of freight rates derive from two assumptions. First, "consumer" behavior is summarized by a demand curve for freight transport. And second, the supply side is characterized by one of a variety of industrial structures, e.g. monopoly, duopoly, perfect competition, etc. This modelling approach gives well known results. For instance, if a carrier has a monopoly in a particular market, shippers will pay the monopoly price, and therefore ship less than what is socially optimal. Accordingly, there is room for a regulatory authority to force the carrier to charge a lower price.

Our approach does not begin with the usual demand curve for transportation. Typically such a demand curve summarizes the requirements of a group of shippers in a particular OD market. Rather, we focus on one shipper. The profit function of this shipper is modelled explicitly, so that his demand for transportation is implicit in his first-order condition for profit maximization. Hence we focus on the way a single shipper contracts with one or more carriers. This allows us to consider the welfare problem in the context of the vertical exchange literature. [1] Necessarily we study a bilevel game where carriers choose a tariffs, and based on these tariffs, a shipper decides how much output to move to market over each carrier.

[1] An excellent introduction to the vertical exchange literature can be found in Tirole (1988).

Not surprisingly, the optimal form of carrier tariff is nonuniform. That is, a carrier can always make a higher profit by charging a nonuniform unit price rather than a constant unit price. In relatively simple vertical structures, a characteristic of this optimal nonuniform price is that *the shipper will choose the same output he would have had the carrier priced a marginal cost.* Thus, relative to Deregulated Prices, economic efficiency is not enhanced by moving to a regime of Marginal Cost Pricing.

A similar argument applies to Ramsey Pricing. In the case where the carrier is forced to use a Ramsey Price which exceeds marginal cost, the shipper will choose a *lower* output than he would have had the carrier been free to choose a profit-maximizing price. Thus the imposition of Ramsey pricing degrades economic efficiency.

2 A Digression on Welfare and Pricing

The argument for Marginal Cost Pricing is well known. In a perfectly competitive economy, prices will be equated with marginal cost and efficiency will be maximized. However, if a market is imperfectly competitive, there is a role for market intervention. If a single imperfect market is viewed in isolation, the usual solution is to force the offending firm(s) to price at marginal cost, and thereby enhance economic efficiency. [2]

The rationale for Ramsey Pricing in freight transport markets, particularly rail freight markets, is based on the nature of a carrier's costs. Suppose a carrier has a production technology exhibiting increasing returns to density or scale. If the carrier is forced to price at marginal cost, it will suffer a financial loss since average cost will be below marginal cost. Thus, if a regulator wished to maximize efficiency by forcing the carrier to price at marginal cost, the carrier would soon be bankrupt. Ramsey prices are intended to solve this problem: they maximize economic efficiency *subject to the constraint that the carrier must breakeven.* In general, Ramsey prices are uniform prices set above marginal cost. They have the property that the carrier's margin as a percentage of price, (P-MC)/P, at the Ramsey Price, varies inversely with the elasticity of transport demand.

[2] If there is more than one imperfect market in an economy, then it may not be optimal to impose marginal cost pricing in these markets. This argument is outlined in the seminal work of Lipsey and Lancaster (1956).

Ramsey pricing has its origin in Frank Ramsey's (1927) seminal paper. Among others, Baumol and Bradford (1970) refined Ramsey's ideas. The suggestion that Ramsey pricing be applied to transportation markets is more recent. For instance, Braeutigam (1979) shows how to construct Ramsey prices where two transportation firms offer differentiated products (service) in the same market. Roberts (1983) gives a good discussion of Ramsey prices in the context of railroad maximum rate control. In the early eighties a number of prominent economists (Baumol (1981), Willig (1981), Goldfeld (1981), Arrow (1981), Moses (1981), Braeutigam (1981) and Wecker (1981)) argued that the application of Ramsey prices to railroad ratemaking, the so-called Inverse Elasticity Rule, offered a number of advantages. Faulhaber and Baumol (1988) comment on the importance of Ramsey pricing:

> Ramsey pricing is a clear example of a principle that derives from the literature and has (recently) achieved a good deal of attention among government agencies. Indeed it casts its shadow on virtually every hearing on price regulation by a federal agency in the United States, and it has apparently arisen in similar circumstances in the United Kingdom, Canada, France, Australia, and New Zealand. [3]

They go on to comment on the economic profession's acceptance of Ramsey prices:

> Eventually, however, frequent reiteration by economists in regulatory proceedings and the profession's general (but not perfectly complete) acceptance of Ramsey pricing as the theoretically correct rule for regulation of the prices of a multiproduct monopoly left an impression on regulators. [4]

One of the notable exceptions to this support of Ramsey pricing is the paper by Tye and Leonard (1983). They argue that, since freight transportation demand elasticities are hard to measure, freight rates based on Ramsey prices would be subject to substantial error, so that Ramsey price regulation may lead to an efficiency loss as a result of measurement error.

[3] See Faulhaber and Baumol (1988), page 594.

[4] Ibid., page 595.

3 A Theory of Deregulated Prices

The theory is motivated by a typical market for bulk commodity transportation. Canadian potash is moved from a small number of Saskatchewan mineheads to a collection of some 5,000 fertilizer plants in the U.S. Midwest by Canada's two major railroads, CN and CP. Here, we refer to a potash mine as a *shipper*, CN and CP as *carriers*, and fertilizer plants in the Midwest as *consumers*. We sometimes refer to this fertilizer market as the *delivered market*.

Hence, consider two carriers competing to move a shipper's freight to consumers at a delivered market. We do not restrict the carriers to have the same costs. Assume that the cost function of the lower cost carrier, termed the LC-carrier, is given by $m_L x_L$, where x_L is the output the shipper moves with the LC-carrier. [5] The cost function of the higher cost carrier, termed the HC-carrier, is $m_H x_H$, where x_H is the output moved with the HC-carrier. In all, the shipper moves $x = x_L + x_H$ to the delivered market.

Consumers are prepared to pay the shipper a price $p_0(x)$ if output x is moved to this market. Hence, shipper revenue is $xp_0(x)$. We assume that $p_0(x)$ is a strictly monotone decreasing function.

The shipper has two costs. $C_0(x_L + x_H)$ is the shipper's cost of manufacturing, and represents all costs to get the product to the plant door. The other cost is what the shipper must pay the carriers to move his product to the delivered market. Let $T_L(x_L)$ be the total amount the LC-carrier charges the shipper to move x_L units of output to the delivered market. We refer to $T_L(x_L)$ as a *tariff* or *contract*. In a similar fashion, we define $T_H(x_H)$.

Given these assumptions, the shipper's profit is given by

$$\begin{aligned}\pi_0(x_L, x_H | T_L, T_H) &= (x_L + x_H)p_0(x_L + x_H) \\ &\quad - C_0(x_L + x_H) - T_L(x_L) - T_H(x_H) \qquad (1)\end{aligned}$$

The LC-carrier's profit is

$$\pi_L(T_L | x_L, x_H) = T_L(x_L) - m_L x_L \qquad (2)$$

and the HC-carrier's profit is

$$\pi_H(T_H | x_L, x_H) = T_H(x_H) - m_H x_H. \qquad (3)$$

[5] This cost function is an incremental cost function. It gives the addition to the LC-carrier's total costs as a result of the shipper using the carrier, holding all other traffic on the LC-carrier's network fixed.

The bilevel structure of the game is as follows. The two carriers form the upper level, each selecting a tariff. The shipper forms the lower level, and given the carrier tariffs, selects an output to move over each carrier. Suppose $(\hat{x}_L, \hat{x}_H)$ solves the shipper's problem.

At the higher level, the LC-shipper solves

$$\max_{T_L} \pi_L(T_L|\hat{x}_L, \hat{x}_H), \tag{4}$$

and the HC-carrier solves

$$\max_{T_H} \pi_H(T_H|\hat{x}_L, \hat{x}_H), \tag{5}$$

and these problems are solved simultaneously. We label this game $\mathcal{G}(2C, 1S)$ and suppose an equilibrium for it is $(x_L^B, x_H^B, T_L^B, T_H^B)$.

Before identifying the equilibrium, we need to define several surplus functions. Let $\Gamma(x_L, x_H)$ be the vertically integrated profit function of all three players:

$$\Gamma(x_L, x_H) = (x_L + x_H)p_0(x_L + x_H) - C_0(x_L + x_H) - m_L x_L - m_H x_H. \tag{6}$$

We assume that Γ is maximized at $(x_L^*, x_H^*) \geq 0$, where $\Gamma^* = \Gamma(x_L^*, x_H^*) > 0$.

The vertically integrated profit of the LC-carrier and shipper, assuming the HC-carrier moves 0 output, is $\Gamma(x_L, 0)$. We assume that $\Gamma(x_L, 0)$ is maximized by $\bar{x}_L^* > 0$ where $\Gamma_L^* = \Gamma(\bar{x}_L^*, 0)$. In a similar way we define the vertically integrated profit of the HC-carrier, assuming the LC-carrier moves 0 output. We assume that $\Gamma(0, x_H)$ is maximized at $\bar{x}_H^* > 0$ where $\Gamma_H^* = \Gamma(0, \bar{x}_H^*) > 0$.

We are now in a position to give the equilibrium.

Proposition 1 *Suppose $m_L < m_H$. Then an equilibrium for $\mathcal{G}(2C, 1S)$ is*

$$\begin{aligned} x_L^B &= x_L^* \\ x_H^B &= 0 \\ T_L^B(x_L) &= \begin{cases} \Gamma^* - \Gamma_H^* + m_L x_L & \textit{if } x_L > 0 \\ 0 & \textit{if } x_L = 0 \end{cases} \\ T_H^B(x_H) &= m_H x_H. \end{aligned} \tag{7}$$

Proof

Consider the case where the carriers have generalized cost functions, $C_L(x_L)$ and $C_H(x_H)$, and suppose that an optimal form of carrier tariff is $T_i(x) = b_i + C_i(x)$ for $i = L, H$. Given these tariffs, the shipper has 3 alternatives: he can use both carrier tariffs, or only the LC-carrier's tariff, or only the HC-carrier's tariff.

Suppose the shipper chooses to use both tariffs. Then we can write the shipper's profit function as

$$\pi_0(x_L, x_H) = \Gamma(x_L, x_H) - b_L - b_H. \tag{8}$$

and, if b_L and b_H are sufficiently small, the shipper will choose (x_L^*, x_H^*) since, by definition, this point maximizes Γ. At this output profits are

$$\begin{aligned} \pi_0(x_L^*, x_H^*) &= \Gamma^* - b_L - b_H \\ \pi_L(x_L^*) &= b_L \\ \pi_H(x_H^*) &= b_H \end{aligned} \tag{9}$$

Now suppose b_L and b_H are such that it is optimal for the shipper to choose only the LC-carrier. Then shipper profit is

$$\pi_0(x_L, x_H) = \Gamma(x_L, 0) - b_L, \tag{10}$$

so the shipper chooses $\bar{x}_L^*$, and profits are

$$\begin{aligned} \pi_0(\bar{x}_L^*, 0) &= \Gamma_L^* - b_L \\ \pi_L(\bar{x}_L^*) &= b_L \end{aligned} \tag{11}$$

The final possibility is that the shipper chooses to use only the HC-carrier. In this case, output is $\bar{x}_H^*$ and profits are

$$\begin{aligned} \pi_0(0, \bar{x}_H^*) &= \Gamma_H^* - b_H \\ \pi_H(\bar{x}_H^*) &= b_H. \end{aligned} \tag{12}$$

We now examine the conditions under which the shipper will choose to use both carriers. This will happen if

$$\pi_0(x_L^*, x_H^*) > \pi_0(\bar{x}_L^*, 0) \text{ and } \pi_0(x_L^*, x_H^*) > \pi_0(0, \bar{x}_L^*) \tag{13}$$

or if

$$\begin{aligned} b_L &< \Gamma^* - \Gamma^*_H \\ b_H &< \Gamma^* - \Gamma^*_L. \end{aligned} \tag{14}$$

(There is an indifference point when one or both these conditions hold with equality.) The shipper will choose only the LC-carrier if

$$\begin{aligned} b_H &> \Gamma^* - \Gamma^*_L \\ b_L - b_H &< \Gamma^*_L - \Gamma^*_H. \end{aligned} \tag{15}$$

Finally, the shipper will choose only the HC-carrier if

$$\begin{aligned} b_L &> \Gamma^* - \Gamma^*_H \\ b_L - b_H &< \Gamma^*_L - \Gamma^*_H. \end{aligned} \tag{16}$$

We now claim that the carriers will choose

$$\begin{aligned} b_L = b^B_L = \Gamma^* - \Gamma^*_H \\ b_H = b^B_H = \Gamma^* - \Gamma^*_L. \end{aligned} \tag{17}$$

To see this, suppose the LC-carrier chooses b^B_L and the HC-carrier chooses $b_H < b^B_H$. Then the HC-carrier can increase profit by raising b_H to b^B_H since the shipper will continue to choose both carriers. A similar argument shows that the LC-carrier will never choose $b_L < b^B_L$. Now suppose the HC-carrier chooses an output $b_H > b^B_H$. Then the LC-carrier can choose $b_L = b_H + \Gamma^*_H - \Gamma^*_L - \epsilon$, where $\epsilon > 0$, and at b_L, the shipper will choose to move 0 output over the HC-carrier. Thus HC-carrier profit will go to 0. Thus the HC-carrier will not choose $b_H > b^B_H$. A similar argument can be made for the LC-carrier.

Now we argue the optimality of the assumed form of carrier price. The question is whether the LC-carrier can find another from of price scheme which increases profit, holding the form of the HC-carrier's fixed at $b_H + C_H(x)$. Suppose there is such a scheme and that this scheme gives the LC-carrier profit of

$$\pi_L > \Gamma^* - \Gamma^*_H \tag{18}$$

Whatever this price scheme, the shipper must make profits at least those he would make if he used only the HC-carrier, or

$$\pi_0 \geq \Gamma^*_L + \Gamma^*_H - \Gamma^*. \tag{19}$$

The implication of (18) and (19) is that

$$\pi_0 + \pi_L > \Gamma^*_L \tag{20}$$

which is impossible since $\pi_0 + \pi_L$ can be no greater than Γ^*_L by definition.

Thus there is no price scheme which will make the LC-carrier's profit higher, and $b^B_L + C_L(x)$ is an optimal form for the LC-carrier's price. Along similar lines, we argue the optimality of the HC-carrier's price.

Having completed the argument for the general case, we now argue the proof for the cost functions $C_L(x_L) = m_L x_L$ and $C_H(x_H) = m_H x_H$. Note that, since $m_L < m_H$, $\Gamma^* = \Gamma^*_L$. Therefore $b^B_H = 0$. Examining the shipper's first-order condition for profit maximization gives $x^B_L = x^*_L$ and $x^B_H = 0$. Consequently $b^B_L = \Gamma^* - \Gamma^*_H$.■

Thus, in a regime of commercial freedom, carriers offer nonuniform prices which cause the shipper to choose outputs, x^*_L and x^*_H, which maximize the joint profit of all three agents (shipper and both carriers). In the case where carrier cost functions are linear and different, the shipper will contract only with the lower cost carrier.

In the proof of the proposition we showed that an optimal form of price is a nonuniform cost-plus price. For linear carrier costs this optimal price is a two-part tariff. However in the rail freight markets we are familiar with, the nonuniform price of choice is a *quantity-forcing* price which takes the form

$$T_Q(x) = \begin{cases} (p+\Delta)x & \text{if } x < \bar{x} \\ px & \text{if } x \geq \bar{x} \end{cases} \tag{21}$$

where $\Delta > 0$. The following proposition, offered without proof, demonstrates that a quantity-forcing tariff can be an equilibrium tariff for $\mathcal{G}(2C, 1S)$

Proposition 2 *Suppose $m_L < m_H$. Then an equilibrium for $\mathcal{G}(2C, 1S)$ is*

$$\begin{aligned} x^B_L &= x^*_L \\ x^B_H &= 0 \\ T^B_L(x_L) &= \begin{cases} (m_L + u^* + \Delta)x_L & \textit{if } x_L < x^*_L \\ (m_L + u^*)x_L & \textit{if } x \geq x^*_L \end{cases} \\ T^B_H(x_H) &= m_H x_H \end{aligned} \tag{22}$$

where Δ is a large positive number and $u^* = (\Gamma^* - \Gamma^*_H)/x^*_L$.

In the next section we compare this Deregulated Price equilibrium with Marginal Cost Pricing and Ramsey Pricing.

4 Regulated Prices

For the game defined, the standard measure of *economic efficiency* is

$$W(x_L, x_H) = \int_0^{x_L+x_H} p_0(z)dz - C_0(x_L + x_H) - m_L x_L - m_H x_H \tag{23}$$

where the first term measures consumer surplus in the delivered market. The variable of interest is *output*, which, in this case, is the vector (x_L, x_H). However, when $m_L < m_H$, $W(x_L, x_H)$ is maximized at $x_H = 0$. Therefore, given the specific carrier cost functions we have assumed, $W(x_L, x_H)$ increases as x_L increases. Hence, to compare the options under consideration – MCP, RP, and DP – we need only compare on the basis of which gives the highest value for x_L.

First, we compare MCP and DP. We assume that each carrier is forced to offer a constant unit price of transportation equal to mariginal cost. Thus the LC-carrier offers m_L, and the HC-carrier, m_H.

Lemma 1 *If each carrier prices at marginal cost, the shipper will choose outputs*

$$\begin{aligned} x_L &= x_L^* \\ x_H &= 0 \end{aligned} \tag{24}$$

Proof
Given marginal cost pricing, the shippers profit function is

$$\begin{aligned} \pi_0(x_L, x_H) &= (x_L + x_H)p_0(x_L + x_H) - C_0(x_L + x_H) - m_L x_L - m_H x_H \\ &= \Gamma(x_L, x_H) \end{aligned} \tag{25}$$

Hence, the shipper chooses the output which maximizes $\Gamma(x_L, x_H)$, which is $(x_L^*, 0)$. ■

Thus, with Marginal Cost Pricing, the shipper chooses the same output he does under Deregulated Pricing. Each regime provides the same welfare.

Therefore the imposition of Marginal Cost Pricing will not enhance economic efficiency.

Now consider Ramsey pricing. Suppose the regulatory authority, for whatever reason, fixes a unit price of transportation, p^R, which exceeds the marginal cost of the LC-carrier, m_L.[6] Then, letting (x_L^R, x_H^R) be the resulting output, it is easy to show that

$$x^R = x_L^R + x_H^R < x_L^*. \tag{26}$$

That is, under Ramsey Pricing the shipper will choose a lower output. Hence, Ramsey Pricing will lower economic efficiency relative to Deregulated Pricing.

5 Conclusions

We have argued that Marignal Cost Pricing and Ramsey Pricing in freight transport markets cannot enhance efficiency. The rationale is that, relative to a regime of Deregulated Pricing, neither Marginal Cost Pricing nor Deregulated Pricing influence a shipper to choose a higher output. The key to these results is that a carrier's optimal nonuniform price has the characteristic that a shipper will pick the same output had the carrier priced at marginal cost. It appears that the economic rationale for regulating freight rates must appeal to fairness.

References

[1] ARROW K. (1981) "Verified Statement On behalf of the Eastern railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide*. 11 May.

[2] BRAEUTIGAM R.R. (1979) "Optimal Pricing with Intermodal Competition", *American Economic Review* 69, 38-49.

[3] BRAUETIGAM R.R. (1981) "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide*. 11 May.

[6] In theory, our choice of carrier carrier cost functions does not require Ramsey pricing. However we could simply add a fixed component to each carrier cost function and the "conditions" for Ramsey pricing would be met.

[4] BAUMOL W.J. (1981) "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide.* 11 May.

[5] BAUMOL W.J. and BRADFORD D.F. (1970) "Optimal Departures from Marginal Cost Pricing", *American Economic Review* 60, 265-283.

[6] FAULHABER G.F. and BAUMOL W.J. (1988) "Economists as Innovators", *The Journal of Economic Literature* 26, 577-600.

[7] GOLDFELD S. "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide.* 11 May.

[8] LIPSEY, R.G. and LANCASTER, K. (1956) "The General Theory of the Second Best", *Review of Economic Studies* 24, 11-32.

[9] MOSES L.N. "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide.* 11 May.

[10] RAMSEY F.L. (1927) "A Contribution to the Theory of Taxation", *Economic Journal* 37, 47-61.

[11] ROBERTS M.J. (1983) "Railroad Maximum Rate and Discrimination Control", *Transprt J* 22, 23-33.

[12] TIROLE, J., *The Theory of Industrial Organization*, MIT Press, Boston, (1988)

[13] TYE W.B. and LEONARD H.B. (1983) "On the Problems of Applying Ramsey Pricing to the Railroad Industry with Uncertain Demand Elasticities" *Transportation Research* 17A, 439-450.

[14] WECKER W.E. (1981) "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide.* 11 May.

[15] WILLIG R.D. (1978) "Pareto-Superior Nonlinear Outlay Schedules", *Bell Journal of Economics* 9, 56-69.

[16] WILLIG R.D. (1981) "Verified Statement On Behalf of the Eastern Railroads in Ex Parte No. 347 (Sub No. 1)", *Coal Rate Guidelines - Nationwide.* 11 May.

GREEK COASTAL SHIPPING SYSTEM: IMPACT OF MARKET DEREGULATION AND NEW TECHNOLOGIES ON MODAL SPLIT[1]

Harilaos N. Psaraftis, National Technical University of Athens
George J. Nellas, Attiko Metro AE
Vangelis Magirou, Athens University of Economics
George C. Nassos, National Technical University of Athens

ABSTRACT

The purpose of this paper is to investigate the problem of modal split for passengers and vehicles in a specific context, that of the Greek coastal shipping system. The transport modes considered are conventional passenger/car ferries (P/C vessels), fast (30-50 knot) vessels, and air transport. For a variety of reasons, monumental changes are about to take place within this system over the next decade. These center primarily on the deregulation of the market that is a result of the European Union integration, and on the introduction of vessels capable of carrying passengers and cars at high speeds. By EU directive, the Greek coastal market shall be fully deregulated by the year 2004. This means that owners would be able to set up routes with minimal governmental interference. The question is of course how passenger demand will evolve within such a new environment, and how the various competing modes of transport will fare. This paper is an attempt to systematically analyze scenarios that might be the possible outcomes of these changes.

1. INTRODUCTION

The purpose of this paper is to investigate the problem of modal split for passengers and vehicles in a specific context, that of the Greek coastal shipping system. The transport modes considered are conventional passenger/car ferries, fast (30-50 knot) vessels, and air transport. For a variety of reasons, monumental changes are about to take place within this system over the next decade. These center primarily on the deregulation of the market that is a result of the European Union integration, and on the introduction of vessels capable of carrying passengers and cars at high speeds. This paper is an attempt to systematically analyze scenarios that will be the possible outcomes of these changes.

The Greek coastal shipping system is one of the biggest in Europe. Between 10 and 15 million passenger trips have been generated within this system each year over the last 5 years. The system involves the movement of passengers, vehicles, and freight within a complex network of mainland-to-island, island-to-island, and mainland-to-mainland connections. About 70 islands in the Aegean and Ionian seas are considered important from an economic perspective (although the actual number of islands is on the order of several thousand). These islands and the mainland are

[1] paper presented at the TRISTAN-II conference, Capri, Italy, June 23-28 1994.

served by a total of 138 ports, of which 42 ports are in the mainland and the rest (96) are island ports. Crete is Greece's largest island, with 8 ports.

Within this geographical area, the Greek Ministry of Merchant Marine has established a system of official "lines," linking the above ports with one another. Some lines also extend across the Adriatic to ports in Italy (Brindisi, Ancona, Venice, and Trieste). Each such line is defined not as a fixed sequence of port calls, but rather as a set of geographical clusters that are internally linked by a network of ship routes (e.g., Piraeus- Crete, Piraeus- Cyclades- Dodecanese- Crete, Patras- Corfu- Italy, etc.). There are close to 100 such lines, spanning the entire system, and broken down in 5 major classes: main lines, secondary lines, local lines of the Argosaronikos bay, other local lines, and freight lines. There can be more than one main line serving a specific geographical area (e.g. for the Cyclades there are 10 different lines). Within a specific line, a variety of individual routes can serve the ports that belong to the line. The total number of ship routes within this line system is on the order of several hundred.

If one excludes deepsea cargo vessels that make direct calls to the mainland or the Greek islands from overseas destinations, cruise ships that exclusively cater to the tourist industry, as well as feeder ships that carry freight within the system, there are several major categories of ships that cater to the passenger transport market. In addition, passengers without cars have always the option of using Olympic Airways or Olympic Aviation.

Perhaps the dominant category of ships is the fleet of conventional ferries for the transport of passengers, private cars, buses, motorcycles, and freight-carrying trucks. These ships operate on the line system described earlier. They have virtually displaced the traditional passenger-only coastal ships that provided service to the islands in the 50's and 60's. Such ships still exist, but their numbers are steadily declining. Having lost a significant share of their market to ferries on long-haul routes, these ships seem to be losing the battle on shorter routes too (their main theater of operation today), this time to high-speed vessels, such as hydrofoils, catamarans, etc. The evolution from traditional passenger-only ships to (mixed passenger/ car) ferries is the first significant transformation of the nature of shortsea shipping in Greece in recent years. This transformation has developed over the course of the last 20-30 years and has been spurred to a significant extent by rapid island economic growth and by significant infrastructure improvements in island ports in the late 60's and early 70's. This allowed for the first time ferry service to these islands. Thus, the ferry mode of operation, practically nonexistent in Greek coastal shipping a few decades ago (except for very short distances and for the island of Crete that had adequate port facilities), has steadily grown since the 70's, by extending service to virtually all Aegean and Ionian islands. Due to the introduction of larger and larger ferries, this mode has ballooned to explosive proportions in the last 3-5 years, and constitutes now an integral and the strongest component of the system. There are on the order of 110 large such ships (above 1,000 gross tons) and on the order of 220 smaller ones (between 100 and 1,000 tons) in the system today.

Today, another transformation seems to be slowly taking place in the composition of the fleet. This transformation concerns the emergence of "fast" (30-50 knots) vessels. This category of ships is worthy of investigation because it has the potential to radically transform the nature of the coastal shipping industry in Greece in the years ahead.

An important component of the fast vessel fleet consists of hydrofoils. These have been operating in short-distance routes for the last 10-15 years. These routes mainly serve the islands in the Argosaronikos Bay near Piraeus, although in recent years the operational range of these vessels has been getting longer, with service to several islands in the Aegean, weather permitting (up to about Beaufort 6). As a result of their high speed (on the order of 30 knots) and reasonably reliable service, these vessels have won a significant share of the passenger traffic on these short-distance routes over the traditional displacement ships that were serving these routes in the past. It is estimated that close to 50 hydrofoils are operating today in Greek waters, still increasing their market share against conventional pure passenger ships.

What has however spurred serious discussion on the potential of this general class of vessels for Greece has been recent interest for other, newer types of fast vessels. These other "new technology" types include SES (surface-effect ships), catamarans, and SWATH (small waterplane area twin hull) vessels. Of these types, two SES ships have already begun passenger-only service to some islands in the Central Aegean, reducing traditional 5-7 hour trip times to 2-3 hours. Also a 35-knot catamaran has begun service in the Argosaronikos Bay in parallel with hydrofoils. Such short travel times may put marine transport on a comparable basis with air transport (particularly over shorter distances).

The recent introduction in other European markets (Italy, or the Channel) and elsewhere (Japan) of new fast designs that can also carry vehicles may radically transform the entire picture of shortsea services in Greek coastal waters in the years ahead, should these designs be allowed to operate in Greece. The growing interest in the potential of these vessels has induced a heated debate on their technical, service, and economic merits for shortsea routes in Greece. Adversaries in this debate are the owners of and prospective investors in such vessels and the owners of the conventional shortsea services (ferry and pure passenger), who are afraid of losing market share to the fast vessels. Arbitrator to the debate is the Ministry of Merchant Marine, tasked by national law to issue permits for the operation of vessels, to approve routes, and to set fares for every type of service rendered. However, owners of fast vessels often complain that the current excessive rigidity in regulation is an obstacle to the issuance of permits, and many times suspect that lobbying by the owners of conventional ships is the real reason for such difficulties.

By EU directive, the Greek coastal market shall be fully deregulated by the year 2004. This means that owners would be able to set up routes with minimal governmental interference. In addition, air transport will also become increasingly deregulated in the years ahead. The question is of course how passenger demand will evolve within such a new environment, and how the various competing modes of transport will fare.

This paper attempts to answer this question by examining various scenarios for the following modes of transport:

(1) conventional ferries (passenger/car),
(2) hydrofoils,
(3) other fast vessels (passenger only),
(4) other fast vessels (passenger/car), and
(5) air transport.

The modal split is based on the "logit" model and the "generalized cost" concept. The cost components used are the fares and the time value of the trip. The time values have been derived from a "revealed preference" dataset. The paper describes the various assumptions made in data collection and model formulation, and discusses the results of the analysis and the additional research needed in this field. Based on this analysis, the profitability of each mode is estimated, leading to an assessment of their impact on the entire system. Policy recommendations are finally offered for an improved operation of the system in view of the monumental changes that are about to occur.

The specific contribution of the paper is the application of this methodology to a context that involves passenger and car coastal transport, several transport modes (including air), and a nontrivial network of origins and destinations. To the best of our knowledge, other coastal shipping modal split applications so far have involved either only freight transport, or have dealt with a more limited network configuration or number of transport modes (such as in the case of the Channel tunnel). The application of such an analysis to the Greek shipping system can serve as a model for its application in other geographical areas, either in Europe or elsewhere. It can also help derive important insights as to what is likely to happen as European coastal shipping markets become increasingly deregulated in the years ahead.

This paper is one of the products of a large project on Greek Coastal shipping, carried out by NTUA on behalf of the Hellenic Industrial Development Bank (ETBA) during 1993, and in the context of the SPA programme of the EU (Regional Development Plan). The project, hereinafter referred to as the ETBA project, carried out a comprehensive investigation of all major aspects of the system, including the topic covered here. Complete details can be found in Psaraftis (1993) and in Psaraftis et al (1994), recently presented at the Second European Research Roundtable Conference on Shortsea Shipping in Athens[2].

This paper is structured as follows. Section 2 gives some background on the system. Section 3 performs the modal split analysis. Section 4 provides some information on the economic viability of fast ships. Finally Section 5 makes some concluding remarks and offers some policy recommendations.

2. SYSTEM FEATURES.

The basic characteristics of the Greek coastal shipping system have been described in a previous paper, presented at the First European Research Roundtable Conference on Shortsea Shipping in Delft (Psaraftis and Papanikolaou (1992)). However, as that paper was written both before the ETBA project had started, and, before the passing of the EU Regulation on maritime cabotage (7 December 1992, see below), some of the data and hypotheses presented in that paper are now obsolete. Thus, before we proceed with our analysis, we deem necessary to (re)familiarize the reader on some features of the system, with a focus on these elements that are more relevant for our analysis. The basic reference for this material is the ETBA final report (Psaraftis, 1993), which describes all this in more detail (see also Psaraftis et al (1994)).

[2] The present paper is a condensed version of Psaraftis et al (1994).

1) Lines and routes. The Ministry of Merchant Marine (MMM) classifies the 102 official lines of the network in 5 classes: (a) 16 main passenger/car ferry (P/C) lines, (b) 30 secondary P/C lines, (c) 11 local P/C lines of the Argosaronikos bay, (d) 39 other local P/C lines, and (e) 3 main and 3 secondary freight (ro-ro) lines. Within this "line" system, the number of individual routes and schedules that are traveled is on the order of several hundreds.

Some of these lines extend to ports in Italy (Brindisi, Bari, Ancona, and Trieste), although from a legal standpoint the services to Italy are not subject to internal cabotage legislation (e.g., ships can fly foreign flags, even if Greek-owned).

2) Fares. With the exception of First Class fares, which are in principle free (with a theoretical maximum of 4 times but in practice 2.8 - 3 times the level of the corresponding Third Class fare), all other fares are uniform for all ships and established every year by the MMM for all pairs of ports. Fares include Second Class, Third Class, Tourist Class, and fares for vehicles (cars, buses, trucks, and motorcycles). Hydrofoils and catamarans have special fares for the routes on which they operate, all (still) regulated by the MMM. There are services in which the official fare with or without a cabin is exactly the same, cabins being allotted to passengers on a first-come first-served basis, many times onboard the ship (in which case the tip to the steward plays the role of the fare supplement).

The rule of thumb that the triangle inequality (Fare (A-B) ≤ Fare (A-C) + Fare (C-B)) holds for most of the network seems to be true, but in general there seems to be no consistent logic in the fare structure, nor there exists a well-defined algorithm or procedure for fare determination.

3) Fleet. The mean age of large (1,000 GRT or more) P/C vessels increased by 4 years (to 25) in the 4 years from 1988 to 1992. The situation is worse for the smaller conventional P/C vessels (between 100 and 999 GRT), with a mean age of 28 years, and even worse for the small (100 to 500 GRT) general cargo (feeder) ships, with a mean age of 35 years (in 1992). There is a mandatory withdrawal age of 35 years for P/C ships (which, interestingly enough, does not apply to ships on the Italian service routes). Thus, at 2004, many ships that operate today within the system will have been withdrawn from service.

In 1992, hydrofoils had a mean age of about 15 years, while the three catamarans in the system (one of which was seriously damaged in 1993 and may never again engage in service) were virtually new. Although hydrofoils have been traditionally restricted to protected waters, 1993 saw the deployment of hydrofoils to several new lines, including many of the Central Aegean islands where the sea is sometimes rough during the summer.

4) Passenger and vehicle traffic. With about 12 million passenger movements in 1990 (see Section 3 for estimates in subsequent years), Greek coastal shipping is one of the biggest in Europe. With few exceptions (short periods of temporary decline), passenger traffic has steadily grown every year over the last 30 years, from approximately 3 million movements in 1964, to about 5 million in 1970, 8 million in 1980, and 8.5 million in 1985. There was a period of decline from 1981 to 1983, with a local minimum of 7.5 million.

The heaviest traffic is generated within the short-distance routes of the Argosaronikos system, with traffic that is more than double in passenger movements than that of the long-haul Piraeus- Crete lines. The biggest growth in recent years has been experienced in the Volos-Euvoia-North Sporades lines, mainly due to the massive influx of hydrofoils in that area, and in spite of the decline in conventional vessel passenger traffic that resulted because of this entry.

Vehicle traffic has also grown, in many cases more steeply than passenger traffic. The Piraeus - Crete line is the leader for both cars and trucks, with car movements experiencing a 48% growth between 1981 and 1990, more than double the equivalent passenger growth rate. The introduction of large P/C vessels has been the main reason for the generation of such a demand.

Competing with sea transport of passengers in many mainland and island destinations is air transport, provided by Olympic Airways and its "commuter" subsidiary, Olympic Aviation. Growth between 1980 and 1992 has been mixed, with the peak of about 5.3 million annual trips in 1985, and a lowest level of about 3.2 million trips in 1991 (the year of the Gulf war). A few of these destinations are also served directly by foreign airlines (charter or regular flights).

5) Legal regime. The most significant recent development in the legal arena has been the passing by the Council of the EU of Regulation No. 3577/92 (7 December 1992), regarding the freedom of service in maritime cabotage trades. Such regulation (heretofore referred to as "the Regulation") stipulates, among other things, that Greece's coastal shipping market becomes fully deregulated and open to other EU-flag ships by Jan. 1, 2004. The 11 year waiting period (already reduced to less than 10 years) was intended to provide Greece with the necessary time to prepare for the opening of the market to competition.

Describing the Regulation vis-a-vis the national legal regime, or the probable impacts of the removal of cabotage privileges, or finally what should be done to prepare for 2004, is beyond the scope of this paper. The ETBA final report (Psaraftis, 1993, section 3.6) and Sturmey et al (1994) deal with these issues in more detail. However, as the adoption of the Regulation is the actual reason behind the analyses reported in our paper, we shall be referring to it and to some of its provisions whenever this is necessary during the course of this paper.

With these preliminary considerations, we now proceed with our analyses.

3. MODAL SPLIT ANALYSIS

In the summer of 1993, the Italian company Tirrenia Navigazione introduced the fast monohull GUIZZO in the line between Civitavecchia (mainland Italy) and Olbia (island of Sardinia). The GUIZZO, built by Rodriquez Aquastrada, is a state-of-the-art fast ship, capable of carrying 450 passengers and 126 cars at speeds up to 43 knots. The trip (124 nautical miles) is traveled in 3.5 hours, of which 3 hours are at the maximum speed. Two daily trips were planned for the summer high season, dropping to one at lower traffic seasons. The GUIZZO was scheduled to operate only 11 weeks per year (July- October), and charged for cars a fare only 15% over the equivalent conventional fare.

Such a low high-speed supplement is also charged by the wave-piercer catamarans (such as the HOVERSPEED GREAT BRITAIN) that cross the Channel. Both cases, although completely different in terms of vessel design, enjoy remarkable capacity utilization rates, being generally preferred by the public over the conventional, slower ferries.

In view of the EU Regulation, the appearance of such ships in Greece is considered only a matter of time. Note that as today in Greece there are no fast vessels that can also carry vehicles, conventional P/C ships have a real monopoly on those passengers who travel with their cars (captive demand). The rest of the fast ships operating today are hydrofoils and catamarans, neither of which can carry cars. And although hydrofoils have carved their own special niche in the market, catamarans have been less successful. Technical factors such as sea worthiness have probably little to do with this state of affairs (other than a catamaran collision with a pier in 1993). Their meager presence is mostly attributed to the existing system of route licensing, which, in one case, granted a license to a catamaran on the condition that it serve a 10- port route. It is obvious that such a condition anihilates any speed advantage of these ships over conventional ships and makes their operation uneconomic.

Since the EU Regulation presumably will make route licensing more rational, a natural question to ask is what portion of passenger demand will shift to fast ships (including fast ferries), when these, in fact, are permitted to operate within the system. Given that the passengers would be able to choose among several competing modes, what will be the modal split? It is the purpose of this section to try to answer this question. Note that by "mode" here we mean not only the general distinction between sea and air, but also the finer grain distinction among the various types of vessels (more on this later).

Another (albeit related) question is what is the economic viability of these fast vessels. This question is addressed in Section 4.

Performing the modal split analysis is by no means an easy task, for a number of reasons. First, the coastal shipping network in Greece is huge (138 ports, 34 airports, thousands of inter-port links). Second, one has little or no idea of what will actually happen during the 10 years to 2004 in terms of the fleet, introduction of new technologies, port expansion, and development of legislation, to mention just a few of the crucial factors. Third, it is not immediately clear how the Greek traveler values his or her time, which is perhaps the most critical parameter that one needs to know in order to assess how much more the traveler is willing to pay in order to travel faster.

Some additional difficulties exist (for instance, lack of origin-destination (O-D) flow data). These difficulties will be described in the course of the exposition that follows. Last, but not least, we are aware of no similar analyses in other coastal shipping problems that involve such difficulties. Most of the analyses involve freight (for which the issue of fast transport is different), and/or much simpler network configurations (for instance, the analysis for the Channel Tunnel).

In the face of this complex situation, the approach that we adopted consists of the following steps:

STEP 1: Choose a workable (but hopefully relevant) subset of the entire network for the analysis.

STEP 2: Make aggregate demand projections on this network up to 2004.

STEP 3: Make some assumptions on what kinds of transport modes provide service on this network, and for each evaluate the transit times for the relevant links of the network.

STEP 4: Make some assumptions on the fares charged by each mode.

STEP 5: Calculate the monetary value of the time of the passengers.

STEP 6: Run the logit model to determine the modal split on each branch of the network.

STEP 7: Interpret results and perform sensitivity analysis.

The main advantage of such an approach is that it bypasses the problem of trying to predict inherently unpredictable scenarios, and produces a flexible tool, by which "what if" assessment of scenarios can be performed. Such a tool can readily be applied to larger networks and alternative scenarios (not only for Greece) once the appropriate data have been assembled.

We now describe the work involved in each of these steps, bearing in mind that the complete detailed analysis is reported in Psaraftis (1993) (and, to a lesser extent, in Psaraftis et al (1994)).

STEP 1: Choose a workable (but hopefully relevant) subset of the entire network for the analysis.

In making such a choice, the following conditions must be satisfied:

a) There should be a correspondence between ports and airports, so that a comparison between sea and air transport is meaningful.

b) The range of distances between network nodes should be relatively broad.

c) The selected sub-network should represent a non-trivial part of the entire network in terms of traffic volume.

In this vein, we have decided to examine a 9-port, 6-airport network, distributed in 6 geographical "zones" as follows:

Zone	Region	Ports	Airports
11	Attiki	Piraeus, Rafina	Elliniko
21	Mykonos	Mykonos	Mykonos
31	Santorini	Thira	Thira
41	West Crete	Souda, Rethymno	Hania
42	Iraklio	Iraklio	Iraklio
43	Lasithi	Ag. Nikolaos, Sitia	Sitia

Notice first that each zone has at least one port (and sometimes two), and one airport. So condition (a) above is satisfied. Also, inter-zone distances for this network range from 69 nautical miles (nm) (between zones 31-42) to 221 nm (between zones 11-43). So the range of distances is indeed broad.

In terms of size, and even though 9 ports is only a small fraction of the 138 ports in the system, in 1990 total passenger traffic among the 9 selected ports was 19.2% of total Greek coastal traffic.

Also in 1990, total traffic among the 6 selected airports was 27.3% of total Greek domestic air traffic. So from this perspective the selected sub-network is certainly non-trivial.

STEP 2: Make aggregate demand projections on this network up to 2004.

By "aggregate demand" we mean that at this stage we shall not break down demand by mode, ie how many passengers will go by fast ships, how many by air, etc. This will be done later (Step 6). On the other hand, we want to take full advantage of existing data regarding flows of passengers in the network, including the choice of mode made by these passengers.

Before we proceed, and as an aside to our analysis, we state that in Psaraftis (1993), a projection of total passenger demand for sea transport on the entire network and up to year 2010 was made. After several regression analyses, it was determined that the best fit to historical data (1964-1989) is the one described by the following equation:

$$TOTAL\_PAX = \exp(1.271 + 0.0414*(Y-1963)),$$

where TOTAL_PAX is the total passenger trips by sea in year Y. The R**2 of this equation is 0.95, and the t-statistic on the coefficient of 0.0414 is 21.06, both acceptable.

The above equation projects about 16.5 million trips in year 2000, about 19.5 million trips in 2004, and about 25.5 million trips in 2010.

Returning now to Step 2, this step involves two sub-steps. First, create origin-destination (O-D) tables for this network for a number of years in the past, and second, use these to forecast origin-to-destination demand on the network up to 2004.

Creating the O-D tables for the sub-network was a rather tricky task. The first difficulty was that no such data was directly available in the databases of MMM's Statistical Service or anywhere else (as much as a lot of other data was available). To circumvent this problem, the direct assistance of this service was requested, and after a series of estimates on how flows at each port split among different routes, an "expert estimate" of the O-D table of passenger trips by sea in the sub-network for 1990 was finally made (see Table 1). Psaraftis (1993) provides more details on how this table was produced.

Table 1: O-D table for passengers traveling by ship, 1990.

From/To	11	21	31	41	42	43	Total
11		145,879	201,373	357,060	372,855	9,538	1,086,705
21	140,459		28,603				169,062
31	203,281	27,757			14,712		245,750
41	349,526						349,526
42	387,970		11,332				399,302
43	10,890						10,890
Total	1,092,126	173,636	241,308	357,060	387,567	9,538	2,261,235

Doing the same for passenger trips by air in 1990 was far easier, for this data was directly available from Olympic Airways (see Table 2).

Table 2: O-D table for passengers traveling by air, 1990.

From/To	11	21	31	41	42	43	Total
11				148,572	260,554	830	537,119
21	66,231		4,592		1,664		72,847
31	65,466	4,358			2,067		71,891
41	140,226						140,226
42	249,578	1,784	1,940				253,302
43	816						816
Total	522,317	70,000	70,197	148,572	264,285	830	1,076,201

In addition to passengers, O-D tables for vehicles are necessary, for a portion of the total passengers (those who travel with a vehicle) do not have the choice between sea and air transport (captive demand), and these passengers must be identified. Here we assume that a person traveling with a vehicle has already made the decision to do so and thus does not have the choice of taking the airplane (this assumption is true for a truck driver, but may not necessarily be true for a motorcycle driver, a car driver, or a bus passenger, all of whom conceivably can take the plane and use another vehicle at their destination).

Using a similar methodology to the one described for passengers, O-D tables were produced for trucks, buses, cars, and motorcycles traveling in the sub-network in 1990 (these tables are not reproduced here but are available in Psaraftis (1993)).

To estimate now the passengers traveling with these vehicles, an estimate of how many passengers are carried by each vehicle is necessary. We used the estimate made by Martedec S.A. of Piraeus (in the context of a NATO project on Greek coastal shipping) that each truck carries one passenger, each bus 40 passengers, each car 2.5 passengers, and each motorcycle one passenger.

On this basis, it was determined that from all passengers who traveled without a vehicle in the sub-network in 1990, 43% used the airplane and the rest (57%) took the ship. Overall, 68% of the passengers went by ship, and 32% went by plane (see Psaraftis et al (1994) for more details).

Of course, making a projection to 2004 just from 1990 data is impossible, so in principle we need to repeat this procedure for several years prior to 1990. Published coastal shipping data in Greece exists from 1964 on. Unfortunately however, individual route data is not available in a uniform way, and MMM's Statistical Service was unable to provide such information for prior years, as it did for 1990. To circumvent this new obstacle, it was decided to produce some coefficients, which express the data in the 1990 O-D tables as functions of passenger and vehicle flows into the ports of the sub-network. Then we would use these same coefficients to produce the O-D tables from port passenger and vehicle flows in prior years.

Of course, the assumption that these coefficients stay the same is a debatable assumption. However, given that no major changes in the network have occurred in the past, we feel that it is an assumption that can be justified (lacking a better way to proceed).

No similar problem existed for the air transport O-D data, as this was readily available from Olympic Airways for the period of interest.

Having all these O-D tables for the period 1964-1990, the next substep is to project these into the future. A critical assumption here is that the possible introduction of new technology ships within the network in the future will not generate new demand (other than what would be generated anyway, ie even if these ships are not introduced).

This is also a debatable assumption, and one that can be patently false, as demonstrated by several cases in the past (see effect of hydrofoils in the Volos- Euvoia- North Sporades trade, as mentioned earlier). However, counterexamples also exist. In Psaraftis (1993), an analysis of the Argosaronikos system (the heaviest in hydrofoil traffic) in the period 1977-1990 showed that the effect of hydrofoil entry into that market in the mid-seventies was only a shift of demand from conventional ships to hydrofoils, with no documentable generation of new demand. In fact, growth in the above period was only 18% for the Argosaronikos system, as opposed to 111% for the entire network, a clear sign of demand saturation. So in this case hydrofoils did not generate new demand.

Being unable to say whether or not this will be the case for our sub-network, we chose to be conservative and assumed zero generation of new demand because of the possible introduction of fast ships. Of course, our methodology can still be applied if an alternative assumption is used.

Based on this, regression analyses were conducted individually for all inter-zone links of the sub-network, so as to project demand on those links. Projected flows to 2004 are by no means simple multiples of those flows in 1990, as flows in distinct links are projected to grow in a different way (see Psaraftis et al (1994) for details).

In 1990, only two modes of transport were present on the sub-network, conventional P/C vessels (capturing the entire demand of passengers with vehicles and also receiving a share of the demand of passengers without vehicles) and air transport (receiving the rest of the demand of passengers without vehicles).

We are next ready to make some assumptions on the modes of transport that will be available on the sub-network in 2004.

STEP 3: Make some assumptions on what kinds of transport modes provide service on this network, and for each evaluate the transit times for the relevant links of the network.

We assume that a total of five (5) modes of transport will be available in this network in 2004:

Mode 1: Air transport.
Mode 2: Conventional P/C vessels.
Mode 3: Hydrofoils.
Mode 4: Surface effect ships (passenger only).
Mode 5: Fast P/C vessels.

Note first that whereas all modes potentially cater to passengers traveling without a vehicle (those of Table 6), modes 2 and 5 cater only to passengers traveling with a vehicle (those of Table 5).

The second remark is that not all modes are assumed to provide service to every inter-zone link of the network. For instance, it would be unreasonable to assume direct hydrofoil service between Piraeus and Crete, or any type of service between Hania and Iraklio in Crete.

The modes that are assumed to be operational for each link of the sub-network are as follows:

Link 11-21: All modes.
Link 11-31: All modes except mode 3.
Link 11-41: All modes except mode 3.
Link 11-42: Modes 1, 2, and 5.
Link 11-43: Modes 1, 2, and 5.
Link 21-31: All modes.
Link 21-42: Mode 1.
Link 31-42: All modes.

No modes are assumed to operate (at least directly) on other links of the sub-network.

The following additional assumptions have been made:

1) A passenger's trip starts from the time he or she leaves home to the time he or she reaches the trip's ultimate destination (door to door trip).

2) A 30-minute waiting time is uniformly assumed for all modes at both ends of the trip for embarkation and disembarkation.

3) Times from a traveler's home to the port (or airport) of origin and from the port (or airport) of destination to the traveler's ultimate destination have been estimated for each case separately, by making some assumptions on the "centroid" of the location of either end of the trip. The centroid is assumed to be close to the center of the corresponding metropolitan area, and trip times between the centroid and the corresponding port or airport have been calculated separately for each case.

4) To calculate ship transit times, the following average speeds have been assumed: Conventional P/C, 14 knots. Hydrofoil, 30 knots. SES and fast ferry, 40 knots.

Notice that the assumed speed for conventional P/C ship is rather low. This is to reflect the fact that in the existing network of lines, these ships make several stops from zone 11 to zones 21 and 31, and the fact that the trips from zone 11 to zones 41, 42, and 43 are usually made overnight, with an average speed that is very close to the assumed. Overall, the sailing times implied by this speed are very close to the actual ones.

For the fast ships, non-stop services among zones were assumed, and this reflects the speed values assumed.

Inter-zone flight times are given in Table 4 below, and inter-zone sailing distances are given in Table 5 below. Based on these assumptions, it is straightforward to calculate the trip times for all relevant combinations of modes and inter-zone links.

STEP 4: Make some assumptions on the fares charged by each mode.

Full information exists on the fares charged by the two modes that were operational in 1990, for all links of the network served by each. Table 4 shows that in 1990 Olympic Airways had two fare increases (trip times are also shown in that table). Our analysis uses as airfare the average of the three fares that prevailed.

Table 4: Airfares for three periods in 1990 (GRD) and trip times in minutes.

Link	1/1-7/5	8/5-24/9	25/9-31/12	minutes
11--42	8,700	11,200	12,200	45
11--41	7,400	9,500	10,400	45
11--21	6,000	7,700	8,400	45
11--31	7,600	9,700	10,700	55
11--43	11,800	15,100	16,600	85
42--31	5,500	7,100	7,700	40

Table 5 shows the 2nd-class and passenger car fares charged by conventional P/C ships for the various links of the network. All fares are in GRD (1990) and include all relevant taxes and supplements. The last column in Table 5 shows inter-port distances in nautical miles.

Table 8: 2nd class and passenger car conventional P/C fares in 1990 (GRD).

From	To	2nd	Pass.	Distance
Piraeus	Hania	5,080	9,349	146
Piraeus	Rethymno	5,364	9,349	161
Piraeus	Iraklio	5,364	9,349	175
Piraeus	Ag. Nikolaos	6,866	10,765	197
Piraeus	Thira	3,926	12,276	127
Piraeus	Mykonos	3,137	8,970	94

Rafina	Mykonos	2,647	7,366	70
Mykonos	Thira	2,639	7,327	64
Thira	Iraklio	2,326	6,705	69
Thira	Ag. Nikolaos	2,082	8,311	84
Ag. Nikolaos	Sitia			24

Notice that no fares are given between Ag. Nikolaos and Sitia in Crete. This is so because no traffic between these two ports is examined, Sitia's traffic from other ports going through Ag. Nikolaos.

For fares that will be charged in 2004, the following baseline assumptions are made:

1) All mode 1 and mode 2 fares remain constant in 1990 GRD prices.

2) All mode 3, 4, and 5 fares are 15% higher than the equivalent mode 2 fare.

Of course, both sets of assumptions are debatable. In particular, the second assumption may be characterized as not very strong (15% is too low). However, 15% was the increase used by both the GUIZZO and the HOVERSPEED GREAT BRITAIN, so it would be reasonable to want to see what would happen if this were applied to Greece as well. In addition, in Step 7 we shall examine alternative increases and see what happens then.

The assumption of fare constancy (in 1990 terms) in modes 1 and 2 is also debatable, as either of these two modes may decide to adopt a different pricing policy as 2004 approaches. We shall discuss these alternative scenarios and their implications later on.

STEP 5: Calculate the monetary value of the time of the passengers.

How much a passenger values his or her time is a critical factor in the analysis, for this would ultimately determine the traveler's willingness to pay in order to make the trip faster. The relevant question for our problem is whether we can say anything for the value of time of passengers using this particular network.

There are two ways to ascertain somebody's value of time. The first, and generally the best, is the "stated preference" method, in which the traveler answers a detailed questionnaire in order to explicitly define his or her utility function of time versus money. Unfortunately, this method is very expensive and time consuming, and, as such, was not used here.

The second method is the "revealed preference" method, and consists of using historical data on travelers' modal choices in order to draw conclusions on how much the traveler values time.

In Greece, Lioukas (1982, 1993) used a logit model for travelers using rail transport. In his latest study, conducted in the context of the Athens-Piraeus subway system, he derived a value of about 800 GRD per hour (1993 prices).

Of course, it is far from clear whether such a value is applicable for the case of coastal shipping in Greece. In Japan, Akagi (1991) showed a value of time on the order of 3,000 Yen per hour on the average. Obviously, it would be inappropriate to use such a value for our analysis.

The only alternative left was to see if we could derive an appropriate value of time using existing data on the Greek coastal shipping system. As such, we decided to use the 1990 data on the sub-network (Tables 1 and 2), in which there is a clearly revealed preference of those passengers traveling without a vehicle, between air transport and conventional P/C ship.

To use this data, we assume that for a specific trip the travelers' preferences are according to the following multinomial logit model:

$$f_i = \exp(a_i + bp_i + ct_i) / \sum_k \exp(a_k + bp_k + ct_k) \qquad (1)$$

where f_i is the fraction of travelers using mode i, p_i is the fare charged by mode i, t_i is the trip time using mode i, and a_i is the "preference constant" of mode i, reflecting possible natural biases in favor of or against that mode. b and c are the same for all modes, and are both negative.

For two modes i and k, we can see that

$$\ln(f_i/f_k) = \Delta a_{ik} + b\Delta p_{ik} + c\Delta t_{ik} \qquad (2)$$

where $\Delta a_{ik} = a_i - a_k$, $\Delta p_{ik} = p_i - p_k$ and $\Delta t_{ik} = t_i - t_k$.

This expression means that an increase of the fare by one unit can be offset by a reduction of the trip time by b/c. Alternatively, the ratio c/b is the amount the traveler is willing to pay in order to recuce trip time by one unit. Therefore, the value of time we want is the ratio c/b.

A linear regression analysis of (2) with the 1990 data, and with the additional assumption that $\Delta a = 0$ (there is no initial documented bias in favor of either mode) produces the value of c/b = 415 GRD/hr.

It should be noted that the R**2 for this analysis was not that spectacular (0.54), implying that there are probably more factors affecting traveler preference and behavior than those examined by this model (fare and trip time). For instance, it is certainly true that different classes of passengers have different values of time (a businessman who travels by plane has a different value of time from a tourist who enjoys being on the deck of a ship during the entire morning, or from a traveler who enjoys an overnight journey in a cabin). Having no way to measure such differences, we had to settle with the "average" value of time calculated above. We shall use such a value with caution, knowing that it is only an average, and one that probably overestimates the value of time of some travelers (those traveling by ship) and underestimates the value of time of other travelers (those taking the plane).

To validate this model, we applied the value of 415 GRD/hr to the 1990 O-D data to produce what the logit model gives for total passengers traveling without a vehicle and who prefer sea transport for 1990. We then added the passengers captive to sea transport, and produced Table 3. A comparison with Table 1 shows generally acceptable results.

Table 3. Validation of modal split: "predicted" passengers traveling by ship, 1990 (compare with Table 1).

From/To	11	21	31	41	42	43	Total
11		145,767	182,363	341,807	409,816	7,289	1,087,042
21	143,650		22,784				166,433
31	185,167	21,806			11,661		218,634
41	331,072						331,072
42	412,494		9,224				421,718
43	8,229						8,229
Total	1,080,611	167,573	214,371	341,807	421,477	7,289	2,233,129

We finally note that comparing the 415 GRD/hr value with the value of Lioukas (1993), 415 GRD/hr of 1990 are equivalent to about 625 GRD/hr in 1993, which is lower than (although same order of magnitude with) the 800 GRD/hr produced by him.

STEP 6: Run the logit model to determine the modal split on each branch of the network.

Having calibrated the logit model by calculating an appropriate value of time, and having validated it by comparing Table 3 with Table 1, we now run it for 2004 as follows.

First, as to what the value of time will be in 2004, we assume that this will grow (in constant 1990 prices) as the rate of annual growth of Greek gross domestic product. Assuming a 1.5% average growth (in real terms), this value becomes about 510.6 GRD/hr in 1990 prices (unless otherwise noted, all our analysis is expressed in 1990 GRD). This asumption is plausible, for a person will probably value time more if he or she makes more money.

So we examine modal split in 2004 with a value of time equal to 510.6 GRD/hr (1990 prices). Note however that in 2004 the number of possible modal choices in our sub-network is 5 and not 2, as in 1990. Since the value of 510.6 was derived assuming two modes, a question is whether we can use it for the 3 additional modes assumed in 2004. Another question is whether we can use this value for those passengers traveling with a vehicle. Such passengers, having no choice but to use the conventional P/C ship in 1990, have the fast P/C ship as an alternative in 2004.

There is no foolproof way to address either of these two questions. In fact, in a strict sense, the correct answer to both questions is "no," particularly to the second one (somebody traveling with his car will generally have a different value of time from somebody traveling without it). However, the average value of 510.4 GRD/hr is about the only piece of information on travelers preferences we got, and short of scrapping this analysis altogether, we decided to use it in our analysis as best we could. "As best we could" means a number of additional assumptions concerning the way the modal split calculations are made. These are as follows.

a) In 2004 there will be no capacity constraints on the number of available ships or aircraft to meet projected demand on each link of the sub- network.

b) The value of time for all passengers in the system (traveling with or without vehicles) is 510.6 GRD/hr (1990 prices).

c) The fare assumed to be paid by each passenger traveling with a vehicle (those of Table 5) is the second class fare, plus 1/2.5 the corresponding private car fare. This assumption is reasonable for passengers traveling with their private cars (since on the average each car carries 2.5 persons), but neglects possible fare differentiations for bus, truck or motorcycle travelers. These are estimated to be minor. For these passengers, modal split is made between 2 modes, 2 and 5 (binomial logit model).

d) The most important assumption concerns how the modal split should be made for passengers traveling without a vehicle. All 5 modes are present here, and a straightforward way to run the model would be to apply the multinomial logit formula with all 5 modes present, and let the results fall where they may. The initial set of runs were in fact made this way, and showed fast ships and air transport combined capturing from 70% to 88% of total passenger traffic without vehicles if the value of time is 510.6 GRD/hr and if the fast fare surcharge goes from 15% to 100%. If the fast fare surcharge is kept constant at 15%, this combined percentage ranges from 88% to a striking 99.7% of the passenger traffic without vehicles, the latter case (in which conventional ships receiving almost zero passengers without cars) happening if the value of time is tripled. Judging these results as unrealistic, we decided to adopt a different philosophy on how the modal split is made, as follows.

Instead of a multinomial model (split among 5 modes), we used a binomial model in a pairwise sequential fashion. The first split was between air and all ships combined. The second split was between conventional P/C ships and and all fast ships combined. The third split was between hydrofoils and other fast ships combined (SES and fast P/C ships). The fourth split was between SES and fast P/C ships. Notice that each split (except the fourth) is between a distinct single mode and a set of other modes combined. The time and fare parameters of the combined modes were assumed to be those of the one among these modes for which the "generalized fare" (fare plus trip time multiplied by value of time) was the lowest. This is tantamount to assuming that the traveler makes his choice in a sequential fashion, and at each step he or she always compares a mode with the best (in terms of generalized fare) among all other modes still under consideration.

There is no a priori way of telling what selection biases are introduced by this scheme, or whether these biases are systematic. This is so because there is no systematic ranking of the modes according to their generalized fares (as much as there is one according to their trip times and another one according to their fares). However, from the results (and from a comparison with the multinomial logit runs) we speculate that the biases are primarily against the fast ships. In that sense, we consider these runs (coupled with the assumption that the fast ships generate no new additional demand) to be on the conservative side with respect to the future of these ships.

The detailed results of this step are reported in Psaraftis (1993) and Psaraftis et al (1994). Step 7 interprets these results.

STEP 7: Interpret results and perform sensitivity analysis.

As the results concern only a limited application of modal split (sub-network and not entire network), they should be interpreted with caution. For instance, the percentages of each mode depend not only on passenger preferences, but also on our very assumption on what links of the subnetwork are served by each mode. So these results should only be considered an output of a "what if" analysis, and not as predictions of what will actually happen in 2004. At the same time, we consider useful to perform a sensitivity analysis on some of the parameters so as to obtain some additional insights. Sensitivity analysis concerns two main parameters: The fare differential between conventional and fast ships (assumed in the baseline scenario at 15%), and the value of time (assumed in the baseline scenario equal to 510.6 1990 GRD/hr).

In 1990, of those passengers who traveled in the sub-network without a vehicle, 43% traveled by air, and the rest (57%) by conventional P/C ship. In total, 68% took the ship, and 32% used the plane.

In 2004, for those who will travel without a vehicle, 32% will take the plane, 40% will go by conventional P/C ship, 3.3% will take the hydrofoil, 3.7% will use SES, and 21% will go by fast P/C ship. For those who will travel with a vehicle, 60% will go by conventional P/C, while 40% will go by fast P/C.

These percentages, if interpreted narrowly, may be misleading. For instance, for passengers who travel without vehicles, the small hydrofoil and SES percentages (as compared to that of the fast ferries) are mostly due to our assumption on what links of the subnetwork are served by these modes and less on actual preferences. In fact, SES and fast P/C have the same speed and charge the same fare, so on one should expect a tie of these modes on the links served by both. This happens indeed (see Psaraftis et al (1994)). However, not all links are served by both modes, by our own assumption, and that is why the overall shares of mode 5 are higher than those of mode 4.

In addition, these percentages do not differentiate between short and long-haul routes. If we are more careful, we can see that hydrofoils raise their percentage on short-haul routes and other new technology ships do so for longer-haul routes.

The general observation from these runs is that the overall percentage of traffic that goes to the new technology ships (modes 3, 4 and 5) can be significant. This is mainly against the airplane for passengers without cars and against conventional ferries for passengers with cars. One possible reason for this is the small (15%) fast fare surcharge assumed. Irrespective of whether these ships can survive on such a small fare (this will be examined in Section 4), one natural question is what happens to modal split if the fast fares become higher.

To investigate this, we examine what happens if the fast ship fare is 30%, and 50% over the conventional one (ceteris paribus). The results are again differentiated between passengers without vehicles, and passengers with vehicles:

For the former pasenger category, if the fast fare surcharge is 30% (50%) the shares of each mode become: Air, increase to 34% (36%); conventional ferry, slight increase to 41% (41%); hydrofoil, decrease to 2% (1.9%); SES, decrease to 3.1% (2.8%), and fast ferry, decrease to 19.9% (18.3%). For pasengers traveling with a vehicle, the share of the conventional ferry increases to 64% (68%), while that of the fast ferry goes down to 36% (32%). In other words, the main beneficiary of a more expensive fast ship fare is the airplane for passengers traveling without a car and the conventional ship for passengers traveling with a car.

We next examine what happens if the value of time is twice or three times what was originally assumed (with a 15% fast fare surcharge).

For passengers without cars, if the value of time is doubled (tripled), the new shares are: Air, increased to 35% (37%); conventional ships, decreased to 36% (31%); hydrofoil, decreased to 2.4% (2.6%); SES, decreased to 3.6% (3.4%); and fast ferry, increased to 23% (25.4%). For passengers with cars, the shares are: Conventional ferries, dropped to 55% (49%), while fast ferries increase their share to 45% (51%).

We see that if the value of time increases, for both passenger classes the main loser is the conventional ferry, while the main beneficiary is the fast ferry and the airplane. Interestingly enough, the other two fast ship modes see their shares slightly decrease.

4. ECONOMIC FEASIBILITY ANALYSIS

In view of the promising results of the previous section with respect to the possible share of passenger demand that new technology ships might be able to attract in 2004, a pertinent question is what is the economic potential of these vessels. Clearly, a modal split analysis would be incomplete if the economic viability of these vessels is not also assessed. Although such an analysis is not the central focus of this paper (see Psaraftis (1993) for complete details), we provide here a summary of its main results.

The project team collected (and/or estimated) technical and economic data (not reproduced here) for the following categories of new technology vessels:

1) The fast monohull GUIZZO (mainland Italy - Sardinia).
2) The swath AEGEAN QUEEN (under design at NTUA- see Papanikolaou et al (1991)).
3) The wave-piercer catamaran HOVERSPEED GREAT BRITAIN (Channel service).
4) The swath PATRIA (Tenerife service).
5) The SES CORSAIR 900 (under construction in Germany)
6) The hydrofoil KOMETA (in service in Greece).

Of these, vessels 1, 2, 3, and 5 can carry cars, while vessels 4 and 6 can only carry passengers.

A parametric analysis was performed on two important parameters: The vessel's capacity utilization (ranging from 30% to 70%, with 60% assumed as the baseline value), and the company's required return on investment (ranging from 0 to 40%, with 20% assumed as the baseline value).

The vessel's economic performance depends not only on the above parameters, but also on the route it serves, as well as the operating scenario for that route. For instance, if the MMM imposes a mandatory requirement of provision of year-round service, the ship would have to collect higher fares to stay viable than if no such requirement were imposed. So we formulated seven possible scenarios, the following:

Scenario a: Route Piraeus - Mykonos (94 nm), 2 roundtrips per day for the 3 summer months, 1 roundtrip per day for 8 months, 1 month out of service.

Scenario b: Same as scenario a, but 2 roundtrips per day for 11 months, and 1 month out of service.

Scenario c: Same as scenario a, but route is Piraeus - Santorini (126 nm).

Scenario d: Same as scenario b, but route is Piraeus - Santorini.

Scenario e: Same as scenario a, but route is Piraeus - Iraklio (175 nm).

Scenario f: Same as scenario b, but route is Piraeus- Iraklio.

Scenario g: Same as scenario e, but 1 daily roundtrip for 11 months and 1 month out of service.

The purpose of scenarios b, d, and f is not so much to examine the performance of these vessels if the two daily roundtrips of the summer are extended during the rest of the year, but to simulate a scenario in which the shipowner can remove his ship from service during the 8 months of the off-season, and employ the ship outside the Greek system. The assumption is that this alternative employment is equivalent in terms of revenue.

We also note that some of these scenarios do not match some of the vessels. For instance, the AEGEAN QUEEN cannot make the two roundtrips to Crete (scenarios e and f), due to lower speed. Similarly, the PATRIA and KOMETA (that do not carry cars) are not examined at all on this route.

There are 34 vessel- scenario combinations. All are shown in Table 6. The table shows two fares for each vessel- scenario combination:

(i) the (minimum) required passenger fare to break even (on a net present value sense) over the ship's lifetime (codenamed RFR, and expressed in 1990 GRD).

(ii) the passenger fare that maximizes revenue, assuming a binomial logit modal split between the vessel and a conventional ferry charging the conventional fare (codenamed MAX, and also expressed in 1990 GRD).

Psaraftis (1993) provides more detail on how both fares are calculated. MAX is obtained by taking the derivative of the logit equation and then iteratively solving a set of non-linear equations. No retaliation is assumed from conventional vessels.

Also shown in the table are the 2nd class conventional vessel fare, and the airfare for each route.

Table 6: Economic performance of vessels for each scenario.

	Scenario:	a	b	c	d	e	f	g
Ship	Fare							
GUIZZO	RFR	10,453	7,477	11,226	8,250	12,408	9,432	14,640
	MAX	2,825	2,825	3,403	3,403	4,668	4,668	4,668
AEGEAN QUEEN	RFR	5,757	3,936	6,011	4,191			7,123
	MAX	2,686	2,686	3,194	3,194			4,316
HOVERSPEED	RFR	8,092	5,521	8,439	5,869	8,973	6,402	10,901
	MAX	2,732	2,732	3,254	3,254	4,449	4,449	4,449
PATRIA	RFR	5,339	3,566	5,497	3,724			
	MAX	2,693	2,693	3,230	3,230			
CORSAIR	RFR	9,723	6,682	10,246	7,145	10,896	7,854	13,177
	MAX	2,825	2,825	3,403	3,403	4,668	4,668	4,668
KOMETA	RFR	5,158	3,575	5,432	3,849			
	MAX	2,590	2,590	3,054	3,054			
2nd class fare		3,137	3,137	3,926	3,926	5,364	5,364	5,364
airfare		10,558	10,558	11,620	11,620	12,550	12,550	12,550

Several remarks can be made from this table. First, and with the possible exception of the PATRIA and the KOMETA, all other vessels require fares considerably higher than both the conventional fare and their own revenue maximizing fare. These fares become prohibitive (compare for instance with airfares) for scenarios a, c, and e, which require the maintenance of a year- round service.

By contrast, if the year- round service requirement is lifted (scenarios b, d, and f), the RFR's drop considerably.

The above scenario assume a 60% utilization and a 20% required return on investment. If the utilization is increased and/or the rate of return is decreased, the RFR's drop somewhat (see Psaraftis (1993 for the full sensibility analysis).

The above results certainly do not paint a particularly rosy picture for the future of fast ships in Greece, and neutralize, to a significant extent, the promising results of the previous section. They boil down to the realization that although fast ships can attract a significant share of passenger traffic if the fares they charge are modest (15% to 50% over the conventional fares), the economic viability of such vessels is likely to be problematic because they need much higher fares to break even. As these fares are often close to the level of air transport fares, very few people would accept them, rendering the overall operation problematic. The deregulation of European air transport is likely to make matters even worse.

Several factors contribute to this outlook, and to the extent that some or all of these factors change, the outlook itself can change for the better. These are the following:

a) Low level of conventional fares.
b) High relative cost of fast ships.

c) Low value of time in Greece.
d) Operating scenario controlled by the MMM.

At the same time, the outlook can get more complicated if the other modes (1 and 2) cease to adopt a "do-nothing" fare policy (as we assumed) but formulate a fare structure that is explicitly designed to make life even more difficult for new technology ships. The analysis of the impications of such policies (which may contain elements of gaming and oligopolistic price equilibrium theory) are left for a future phase of this research.

5. CONCLUDING REMARKS

This paper presented some modal split scenarios for the Greek coastal shipping system, in view of the lifting of cabotage privileges by 2004. All of these scenarios are hypothetical, but we feel they have a substantial degree of realism so as to be able to perform a "what if" analysis of what is likely to happen.

Our analysis would be stronger if a "stated preference" data set were available instead of the "revealed preference" one, for the latter was seen to exhibit some limitations. Also, a broader analysis for a larger part of the network could provide some additional insights.

In terms of policy recommendations, a lot of work needs to be done in the 10 years to 2004, both by the MMM and by private industry, in order to be able to best adapt to the new game that will be played. Many of such recommendations are listed elsewhere (see Psaraftis (1993) and Sturmey et al (1994)). Within the scope of this paper, we feel that the analysis presented supports the following policy recommendations.

1) Put an end to the tightly controlled fare structure, well before the end of 2003, at least for some types of service.

2) For those routes and services that do not belong to the "public service" sector, allow competition and freedom to set routes and fares.

3) The MMM should set up criteria for the determination of which will be the "public service" routes, and on how licenses will be granted for those.

4) Market surveys should be carried out to determine the "stated preference" of travelers. These are essential so as to be able to predict modal split with an acceptable degree of confidence.

REFERENCES.

Akagi, S., 1991. Synthetic Aspects of Transport Economy and Transport Vehicle Performance with Reference to High- Speed Marine Vehicles. Proceedings, FAST 91 Conference, Trondheim.

Lioukas, S., 1982. Travel Modes and the Value of Time in Greece. Journal of Transport Economics and Policy, May.

Lioukas, S., 1993. Study of Athens' Subway System Investment Program. Centre of Economic Research, Athens University of Economics (in Greek).

Papanikolaou, A.D., G. Zaraphonitis, M. Androulakakis, 1991. Preliminary Design of a High-Speed SWATH Passenger Car Ferry. Marine Technology, Vol. 28, 129-141.

Psaraftis, H.N., 1993. Greek Coastal Shipping: Status, Prospects, and Investment Opportunities. Final Report to ETBA (in Greek), December.

Psaraftis, H.N., A.D. Papanikolaou, 1992. Impact of New Technologies on Shortsea Shipping in Greece. Proceedings, First European Research Roundtable Conference on Shortsea Shipping, Delft, November.

Psaraftis, H.N., V.F. Magirou, G.C. Nassos, G.J. Nellas, G. Panagakos, A.D. Papanikolaou, 1994. Modal Split Analysis in Greek Shortsea Passenger/Car Transport. Proceedings, Second European Research Roundtable Conference on Shortsea Shipping, Athens, June.

Sturmey, S., G. Panagakos, H. N. Psaraftis, 1994. Institutional and Socioeconomic Issues in Greek Ferry Services. Paper to be presented at the Second European Research Roundtable Conference of Shortsea Shipping, Athens, June.

A THEORY OF ROAD PRICING FOR URBAN TRANSPORT NETWORKS

Paolo Ferrari
University School of Engineering of Pisa
Via Diotisalvi 2 - 56126 Pisa - Italy

ABSTRACT

This paper presents a theory of road pricing for urban transport networks, which considers it as a tool of traffic engineering intended to assure the equilibrium of the transport system, avoiding that demand tends to exceed system capacity. The paper shows, within the framework of the traditional theory of transport networks, that urban networks can have no equilibrium pattern, and this absence of equilibrium is responsible for severe congestion that afflicts many cities in the world.

However the existence of a network equilibrium can be assured in any case if additional costs are imposed on network links. A method to calculate this road pricing is proposed, and a discussion about the consequences deriving from this imposition is presented. It is argued that in general, in order to make the imposed equilibrium acceptable from the economic and social points of view, often a modification of the network structure becomes necessary, in which road pricing is integrated with other techniques of traffic control and transport planning.

1. Introduction

The issue of road pricing was used for the first time during the early 1920s to illustrate the concept of social cost within the framework of a broad disagreement about the efficiency of private enterprise. Road pricing was not offered as a viable policy to address a specific social concern over eccessive congestion (Morrison, 1986). Over the years, however, as congestion problems became more widespread in cities throughout the world, the interest in road pricing as a policy measure increased, and greater attention was addressed to the relationships between road pricing and equilibrium of transport systems.

In their theory of road networks Beckmann et al. considered the costs of resources actually consumed by users (time, fuel, etc.) as the only components of generalized transport cost, whereas road pricing was thought as a mere transfer of money from road users to society. Starting from this hypothesis, and considering the cost on a link depending only on the flow travelling along it, they showed that the user equilibrium which reaches the social optimum is obtained if one imposes on each link a road pricing equal to the product of the flow travelling along it by the derivative of its cost function (Beckmann et al., 1956).

The hypothesis that road pricing is not a part of transport cost gives rise to a first objection to the traditional economic theory of road pricing, which tends to ignore the distributional effects and focuses instead upon the point that society, as a whole, will benefit from road pricing (Starkie, 1986). As a matter of fact it is easy to verify that the road pricing paid by each user following the standard theory is, on average, greater than the benefit he receives from the reduction of congestion.

A second, and likely more substantial objection to the traditional theory of road pricing, comes from the fact that, whereas on the one hand it tends to reach the social optimum, on the other it concentrates its attention on a very simple supply and demand model (May, 1986), ignoring the effects of road pricing on demand generation and distribution and on modal split (Hills, 1993).

Starting from the 1960s some extensions and refinements of the economic theory of road pricing have been developed, and some researchers have focused their attention particularly on the distributional effects of road pricing (e.g. Glazer, 1981): but the basic theoretical approach has been substantially unchanged since the early 1920s (Morrison, 1986).

This paper faces the problem of road pricing from a substantially different point of view, and considers it as a tool of traffic engineering intended to assure the equilibrium of urban transport systems, avoiding that demand tends to exceed the system capacity.

2. The behaviour of urban network users

Consider a multimodal urban transport network, represented by a graph $G(N,L)$, where N is the set of nodes and L the set of directed links, each of which is used by only one transport mode. Each link has a cost function associated to it, i.e. a relationship between the average of costs perceived by users travelling along the link and the flow vector on all network links. The transport system operates well only if link flows satisfy some capacity constraints, which are of two kinds, as a consequence of the two different definitions that can be given of street and urban freeway capacity: *physical* (or *crude*) *capacity* and *environmental capacity* (Minister of Transport, 1963).

Physical capacity of an urban freeway is the maximum traffic volume that it can bear: this means that the maximum acceptable risk of flow instability is reached when the freeway is travelled by this traffic volume. When traffic volume tends to exceed capacity there is high risk that flow becomes unstable; in this case marked and sudden speed drops take place, while there is a marked reduction of flow rate. In some cases this phenomenon rapidly disappears, and in a little time there is the recovery of normal circulation conditions. In other cases, for completely random reasons, a temporary recovery is followed by new speed drops (Treiterer and Myers, 1974), and so on for a long time period during which the traffic volume that succeds in travelling along the freeway is much less than its capacity: long queues take place and there is a corresponding increase in journey times.

A sequence of intersections is usually distributed along an urban street. Given the control strategies of these intersections, the physical capacity of a *lane group* (Highway Capacity Manual, 1985) that converges on the intersection is the maximum traffic volume on this lane group that can go through the intersection, with a good probability that the queue which originates in it does not clog the upstream intersection. When this clog occurs, traffic comes to a halt; it is possible that the phenomenon exhausts rapidly, particularly if suitable queue management strategies (Quinn, 1992) are used, and in a little time normal circulation conditions take place again. But if the clog remains for a certain time, traffic stops reach other intersections in the network, causing a corresponding increase in journey times.

Thus both freeways and streets have a common characteristic: travel time increases with flow , in more substantial measure in streets than on freeways, keeping in any case finite values as long as traffic volume is below the physical capacity; when the latter is reached, the consequences are widely variable for completely random reasons, and journey times are very dispersed. As the demand generated by various origins, its distribution among destinations, and modal split depend on journey time, the dispersion of the latter causes the dispersion over time of demand characteristics.

Vehicles travelling along urban streets and freeways cause, among other things, noise, air pollution, danger to pedestrian safety. The damage they produce represents the proportion of the generalised transport cost which is transferred from streets to environment. If the damage that environment can tolerate is fixed, it is possible to calculate the maximum traffic volume which can travel along each street of the network, i.e. its environmental capacity. Various methods have been proposed to calculate the environmental capacity, taking into account noise and pedestrian safety (e.g. Minister of Transport, 1963) and air quality (e.g. Waterfield and Hickman, 1982).

Both physical and environmental capacity constraints on urban freeways refer to each link representative of a freeway stretch, and only the flow on each link appears in them. On the contrary capacity constraints in urban streets are relationships, in general linear, among flows on various links that converge on each intersection.

Some nodes of the network are O/D centroids. At a certain time let be N_i the generation capacity of the centroid i, i.e. the number of the individuals that are in i and that can decide to move from it. The pattern of the network utilization, i.e. the demand between the different pairs of centroids and its distribution among the various transport modes and the various paths, is a consequence of choices made by the individuals who belong to the generation capacity of the various centroids.

Let consider an individual who belongs to N_i and is characterized by a vector s_i of socioeconomic attributes. He chooses the destination j of his trip (j could coincide with i, in this case he would decide not to move at the time we are considering), the transport mode and the path he will use to go to j. So each choice alternative is defined by a destination, a transport mode and a path, and is characterized by a vector of attributes which represent the attractiveness of the destination, the comfort and the charges of the mode, and the path characteristics, i.e. journey time, fuel consumption, etc. A vector of attributes completely defines an alternative, so that when an individual makes a choice, he actually draws a vector of attributes from the *choice set* B_i, which is in general different from an individual to another (McFadden, 1975).

An individual associates an utility value to each alternative. This utility is a function $U(x,s_i)$ of the vector x of the alternative attributes and of the vector s_i of socioeconomic attributes of the individual. He draws from the set B_i the vector x_j which maximizes his utility:

$$U(x_j,s_i) = \max_{x\in B_i} U(x,s_i) \qquad (1)$$

The utility associated to a vector $x\in B_i$ varies for random reasons from an individual to another even if they have the same vector s_i, and it is different also for the same individual in different time periods: so it is a random variable. Let consider the sequence τ_1, τ_2... τ_n... of epochs in which the joint distribution function of U, x, s_i is constant. These epochs can be found out easily: e.g. they are the time periods when people go to work early in the morning, or when they go home in the evening, and so on. It can be noted that this joint distribution is a conditional one, given the values that the attributes of alternatives assume in epochs which are different from those of the sequence $\tau_1, \tau_2 \dots \tau_n \dots$: e.g. the probability that an individual receives a given utility from making or not a trip in a given day period depends on the cost of this trip in another day period.

An individual characterized by the vector s_i associates to the vector $x \in B_i$ in any epoch of the sequence $\tau_1 \dots \tau_n \dots$ an utility given by:

$$U(x,s_i) = \bar{U}(x,s_i) + \varepsilon \tag{2}$$

where $\bar{U}(x,s_i)$ is the regression of U on x and s_i, and ε is a random variable with zero mean. If ε is a Gumbell random variable and if the utilities associated to the various alternatives are independent, the probability that the vector x_j is chosen is given by the *logit model*:

$$P_{ij} = \frac{\exp\left[\bar{U}(x_j,s_i)\right]}{\sum\limits_{x \in B_i} \exp\left[\bar{U}(x,s_i)\right]} \tag{3}$$

Let be y, v, w the vectors of attributes of the destination, of the mode and of the path into which the vector x can be partitioned:

$$x = \begin{bmatrix} y \\ v \\ w \end{bmatrix} \tag{4}$$

We assume that the utility regression $\bar{U}$ has the following additively separable form (Domencich and McFadden, 1975):

$$\bar{U}(x,s_i) = \bar{U}_1(y,s_i) + \bar{U}_2(v,s_i) + \bar{U}_3(w,s_i) \tag{5}$$

It follows from Eq.(5) that an individual belonging to N_i, who has chosen to go to the destination j using the t mode, travels along the path from which he expects the maximum utility, i.e. the minimum cost: thus this minimum cost becomes an attribute of the transport mode chosen to go from i to j.

Let assume that all individuals belonging to N_i have the same socioeconomic characteristics. In this case we have:

$$\bar{U}(x_j,s_i) = a_j + \beta\lambda^t_{ij} \tag{6}$$

where a_j is the component of $\bar{U}$ due to the attractiveness attributes of the centroid j, λ^t_{ij} is the average of the costs expected by users on mode t between i and j along the path considered the most convenient by each of them, and β is a scale factor.

Let be T_{ik} the set of transport modes connecting i and j. By substituting the expression (6) into (3), and taking into account that $\lambda^t_{ii} = 0$, we have:

$$P_{ij} = \frac{\exp\left[a_j + \beta\lambda^t_{ij}\right]}{a_i + \sum_{k\neq i} \sum_{t\in T_{ik}} \exp\left[a_k + \beta\lambda^t_{ik}\right]} \tag{7}$$

If the choices of the individuals belonging to N_i are independent, the trips made in the epoch under examination from i to the other centroids are the components of a multinomial random vector. Thus the average D^t_{ij} of the demand from i to j on mode t in this epoch is given by:

$$D^t_{ij} = N_i \frac{\exp\left[a_j + \beta\lambda^t_{ij}\right]}{a_i + \sum_{k\neq i} \sum_{t\in T_{ik}} \exp\left[a_k + \beta\lambda^t_{ik}\right]} \tag{8}$$

whereas the average of the number of individuals that remain in i is:

$$D_i = N_i - \sum_{k\neq i} \sum_{t\in T_{ik}} D_{ik} = N_i \frac{\exp\left[a_i\right]}{a_i + \sum_{k\neq i} \sum_{t\in T_{ik}} \exp\left[a_k + \beta\lambda^t_{ik}\right]} \tag{9}$$

If a_j and λ^t_{ij} $\forall j,t$ are constant during the sequence $\tau_1 \ldots \tau_n \ldots$, the same happens for the probability given by Eq.(7); in this case the succession of the demand values during the sequence is the realization of a stationary process whose average is given by Eqs.(8) and (9). On the other hand a multinomial random variable converges in probability to its average; this means that, if N_i is large, the probability of a considerable shift of the demand from its average is negligible. For this reason we can consider the demand value constant during all the sequence: Eqs.(8) and (9) give the demand function, and its inverse, i.e. the average λ^t_{ij} of the expected costs of the trip between the i,j pair on mode t as a function of all components of the demand, is obtained from Eq.(8) and (9):

$$\lambda^t_{ij} = \frac{1}{\beta}\left[a_j - a_i - \ln D^t_{ij} + \ln\left(N_i - \sum_{k\neq i} \sum_{t\in T_{ik}} D^t_{ik}\right)\right] \tag{10}$$

Until the socioeconomic characteristics and the attractiveness attributes of the various zones of a town do not vary, the a_j values can be considered constant in the successive epochs of the sequence. The same happens for λ^t_{ij} if the link flows are low, because in this case the costs on network links are independent of the flows. On the contrary, if flows are not low, costs depend on them, and it is possible that λ^t_{ij} are not constant. As a matter of fact the individuals that belong to the generation capacity of the various centroids make their choices at a certain epoch on the basis of the expected trip costs whose averages are λ^t_{ij} $\forall i,j$ $\forall t$. These choices give rise to the demand between the various *O/D* pairs and on the various transport modes, and to flows on network links on which the costs the individuals bear during their trips depend. If these costs are different from those the individuals expected, in the successive epoch they will change their choices, giving rise to a new demand, new link flows and new costs on links, from which new trip costs derive, and so on. It is possible that at a certain epoch such a situation is reached that the trip costs the individuals bear coincide or are very close to those they expected: in this case the individuals will repeat their choices in the successive epochs, giving rise to constant values of λ^{t}_{ij} and of demand.

However this happens only if the transport system has an *equilibrium solution*.

3. The equilibrium solution

Let be (' indicates the transpose of a vector or of a matrix):

$w \equiv (i,j),\ i \neq j$ ≡ an ordered pair of centroids

W ≡ the set of the ordered w pairs

n ≡ the number of pairs $w \in W$

S ≡ the numer of centroids

T_{w} ≡ the set of transport modes joining the pair $w \in W$

$T \equiv \bigcup_{w \in W} T_{w}$ ≡ the set of all transport modes of the network

u ≡ the number of elements in T

r_{w} ≡ the number of all transport modes joining $w \in W$

$r \equiv \sum_{w \in W} r_{w}$

d^{t}_{w} ≡ the number of individuals who travel between $i,j \equiv w$ on mode t during an unit time period within the epoch τ under examination; if we assume that the demand D^{t}_{w} given by Eq. (8) is uniformly distributed during τ, d^{t}_{w} is obtained dividing D^{t}_{w} by the duration of τ.

$d^{t} \equiv (\ldots d^{t'}_{w} \ldots)'$ ≡ the vector of demand during the unit time period on mode t between all $w \in W$ which are connected by mode t

$d \equiv (\ldots d^{t'} \ldots)'$ ≡ the vector of demand during the unit time period on all modes $t \in T$ between all $w \in W$. Hence forward we will omit, for the sake of brevity, the words "during the unit of time" when we will refer to vector d and its components.

P^{t}_{w} ≡ the set of paths p joining w on mode t

$P_{w} \equiv \bigcup_{t \in T_{w}} P^{t}_{w}$ ≡ the set of all paths p joining w

$P^{t} \equiv \bigcup_{w \in W} P^{t}_{w}$ ≡ the set of all paths p joining all $w \in W$ on mode $t \in T$

$P \equiv \bigcup_{w \in W} P_{w}$ ≡ the set of all paths p joining all $w \in W$ on all modes $t \in T$

M^{t} ≡ the number of paths $p \in P^{t}$

M ≡ the number of paths $p \in P$

h_{p} ≡ the flow on path $p \in P$

$h^{t} \in \mathbb{R}^{M^{t}}_{+}$ ≡ the vector of flows on all paths $p \in P^{t}$

$h \equiv (\ldots h^{t'} \ldots)' \in \mathbb{R}^{M}_{+}$ ≡ the vector of flows on all paths $p \in P$

A^{t} ≡ the incidence matrix link-paths travelled on mode t

$f^{t} \equiv A^{t} h^{t}$ ≡ the vector of link flows generated by mode t

$f \equiv (\ldots f^{t'} \ldots)'$ ≡ the vector of link flows generated by all modes.

Given:

$$A = \begin{bmatrix} A^{t_1} & & \\ & \ddots & \\ & & A^{t_u} \end{bmatrix}$$

we have $f = Ah$.

Let be B^t the incidence matrix between pairs w and paths travelled on mode t, and

$$B = \begin{bmatrix} B^{t_1} & & \\ & \ddots & \\ & & B^{t_u} \end{bmatrix}$$

we have:

$$Bh - d = 0. \tag{11}$$

Let suppose that the network has v capacity constraints:

$$g_j(f) < 0 \qquad j \in (1,2...v)$$

where $g_j(f)$ are convex functions (in general linear, as it was said in the previous section).

Any vector $\begin{bmatrix} h \\ d \end{bmatrix} \geq 0$ which satisfies Eq.(11) is called *solution vector*. The set:

$$\Omega_1 \equiv \left(\begin{bmatrix} h \\ d \end{bmatrix} \geq 0 \mid Bh\text{-}d = 0,\ f = Ah,\ g_j(f) < 0 \quad \forall j \in (1,2...v) \right)$$

is the *supply* set; it is bounded and convex because $g_j(f)$ are convex, but is not closed because the capacity constraints are strict inequalities.

The set:

$$\Omega_2 \equiv \left(\begin{bmatrix} h \\ d \end{bmatrix} \geq 0 \mid \sum_{\substack{j \neq i \\ t \in T_{ij}}} d^t_{ij} \leq N_i \quad \forall i \in S,\ Bh\text{-}d = 0 \right)$$

is the *demand* set; it is compact and convex.

The set $\Omega \equiv \Omega_1 \cap \Omega_2$ is the set of *feasible solutions*. It is bounded and convex, because both Ω_1 and Ω_2 are bounded and convex, but it is not closed, because in urban networks, whose study is of actual interest, $\Omega_1 \subset \Omega_2$ and thus $\Omega \equiv \Omega_1$. The set of feasible link flow vectors is:

$$\Theta = \left(f = Ah \mid \begin{bmatrix} h \\ d \end{bmatrix} \in \Omega \right)$$

Let be:

$\bar{\Omega}$ ≡ the closure of Ω

$c_i(f)$ ≡ the average of trip costs perceived by users on link $i \in L$, which is continuous function of f in the set $\bar{\Theta} \equiv \left(f = Ah \mid Bh = d,\ \begin{bmatrix} h \\ d \end{bmatrix} \in \bar{\Omega} \right)$

$C_p(h) \equiv \sum_{i \in L} c_i(f)\delta_{ip}$ ≡ the average of trip costs perceived by users on path $p \in P$, where $\delta_{ip} = 1$ if p traverses i, 0 otherwise, which is continuous function of h in $\bar{\Omega}$

λ^t_{ij} ≡ the inverse of the demand function, obtained by substituting in Eq.(10) $D^t_{ij} = d^t_{ij}\tau$

$\lambda^t \equiv (\dots\lambda^t_w\dots)'$

$\lambda \equiv (\dots\lambda^{t'}\dots)'$

A vector $\begin{bmatrix} \bar{h} \\ \bar{d} \end{bmatrix} \in \Omega$ is an *equilibrium solution* if and only if, for every pair w, every mode $t \in T_w$, and every path $p \in P^t_w$ (Dafermos, 1982):

$$
\begin{aligned}
&\bar{h}_p > 0 \Rightarrow C_p(\bar{h}) - \lambda^t_w(\bar{d}) = 0 \\
&\bar{h}_p = 0 \Rightarrow C_p(\bar{h}) - \lambda^t_w(\bar{d}) \geq 0
\end{aligned}
\qquad (12)
$$

The existence of an equilibrium solution is a necessary condition so that transport demand is constant during the successive epochs of the sequence and at the same time capacity constraints are not violated. As a matter of fact conditions (12) state that at equilibrium the average costs on paths joining each pair w on each mode $t \in T_w$, and which are travelled by users, are equal to the average expected cost λ^t_w, and are not greater than the average costs on paths joining the same w pair on the same mode $t \in T_w$ and which are not travelled. If an equilibrium solution does not exist, this means that the characteristics of the network and of utility function are such that, whatever the actual link flow vector, there are always some paths which, when their capacity constraints are in the point of being violated, have costs that are on average less than those of other links travelled by users and joining on the same mode the same w pairs, and less than the costs λ^t_w corresponding to the demand actually on the network. Users tend to travel along these less costly paths and to exceed their capacity. As a matter of fact this behaviour could be avoided only if users divided into two categories: those who travel along the minimum cost paths, and those who, without receiving any benefit, choose to use paths which cost more, or not to make the trip, so that the capacity constraints would be satisfied everywhere; but this is unrealistic. A violation of capacity constraints gives rise, as it was seen in the previous section, to a great dispersion of journey times during the successive epochs of the sequence, thus to large fluctuations of demand, which give a further contribution to violation of capacity constraints. In these conditions crises of the transport system are very frequent.

However it has to be remarked that the existence of an equilibrium solution is not always sufficient so that demand is constant during the sequence and no crisis of the transport system happens. As a matter of fact it is possible that an equilibrium solution is not reached in the course of the evolution of the transport system. Moreover the demand is not actually uniformly distributed during τ; for this reason, and also because traffic flow is a random phenomenon, traffic peaks that produce system failures can happen even if capacity constraints are not violated. Therefore control strategies suitable to avoid these situations are in general necessary even if an equilibrium solution exists.

It can be shown (Ferrari, 1994) that, if capacity constraints are strict inequalities, conditions (12) are equivalent to the following variational inequality:

$$\sum_{\substack{t\in T_w\\ w\in W}} \Big(\sum_{p\in P^t_w} C_p(\bar{h})\cdot(h_p - \bar{h}_p) - \lambda^t_w(\bar{d})\cdot(d^t_w - \bar{d}^t_w)\Big) \geq 0 \qquad (13)$$

where h_p and d^t_w are the components of any vector $\begin{bmatrix}h\\d\end{bmatrix}\in\Omega$. $C_p(h)$ and $\lambda^t_w(d)$ are continuous functions in Ω, but Ω is not closed, so that the variational inequality (13) does not always admit a solution (Kinderleher and Stampacchia, 1980, pp. 13-14). In the case (13) has not solution, there is not equilibrium in the transport system.

4. The theoretical fundation of road pricing

Let be $Z \equiv \bar{\Omega}\cap(\mathbb{R}^{M+r} - \Omega)$ the part of $\bar{\Omega}$ which does not belong to Ω. The set $\bar{\Omega}$ is convex and compact, the functions $C(h)$ and $\lambda(d)$ are continuous in it, so that the variational inequality (13) has always a solution in $\bar{\Omega}$. If this solution does not belong to Z, it is also a solution of (13) in Ω, thus it is an equilibrium solution of the transport system. We can derive from these considerations a method which assures in any case the existence of an equilibrium solution of the transport system, by modifying the cost functions in such a way that the solution of the variational inequality (13) in $\bar{\Omega}$ does not belong to Z.

Let denote by $c^*_i(f)$ the modified cost function on link i, and let be J_i the set of indices j affecting the capacity constraints $g_j(f) < 0$ in which the flow f_i on link i appears. Given a vector $f\in\bar{\Theta}$:

$$f = (f_1 \ldots f_{i-1}, f_i, f_{i+1} \ldots f_a) \qquad (14)$$

we solve with respect to x_j the equations:

$$g_j(f_1 \ldots f_{i-1}, x_j, f_{i+1} \ldots f_a) = 0 \qquad \forall j\in J_i \qquad (15)$$

and obtain $x_j = \Phi_j(\hat{f})$ $\forall j\in J_i$, where $\hat{f} = (f_1 \ldots f_{i-1}, f_{i+1} \ldots f_a)$. We said before that the functions $g_j(f)$ are linear, so that the functions $\Phi_j(\hat{f})$ always exist and are continuous.

Let be $x_i = \min x_j$ $\forall j\in J_i$. In the following the value x_i obtained in this way from a vector f will be affected by the same index as f: so x_i values deriving from f^* and $\bar{f}$ will be denoted by x^*_i and $\bar{x}_i$ respectively.

Consider a positive constant Δ, as little as possible, bounded from below essentially by computational difficulties. Let be $\bar{\lambda}$ a constant such that $\frac{\bar{\lambda}\Delta}{\Delta+1}$ is an upper bound for the set $\Lambda = \Big(\lambda^t_w(d),\ \forall w\in W,\ \forall t\in T_w \mid \begin{bmatrix}h\\d\end{bmatrix}\in Z,\ d^t_w > 0\Big)$. We define:

$$c^*_i(f) = c_i(f_1 \ldots f_{i-1}, 0, f_{i+1} \ldots f_a) + \frac{\bar{\lambda}}{x_i+1} f_i \qquad \text{if } 0 \leq x_i \leq \Delta \qquad (16)$$

$$c^*_i(f) = c_i(f) \quad \text{if } x_i \geq \Delta \text{ and } f_i \leq x_i - \Delta \tag{17}$$

$$c^*_i(f) = c_i(f_1 \dots f_{i-1}, x_i - \Delta, f_{i+1} \dots f_a) + \frac{\bar{\lambda}}{\Delta+1}(f_i - x_i + \Delta) \quad \text{if } x_i \geq \Delta \text{ and } f_i > x_i - \Delta \tag{18}$$

It easy to verify that $c^*_i(f)$ is continuous in $\bar{\Theta}$, since x_i is a continuous function of $\hat{f}$.

Given a vector $\begin{bmatrix} h^* \\ d^* \end{bmatrix} \in Z$ and $f^* = Ah^*$, it follows from the definition of Z that:

$$g_j(f^*) = 0 \quad \exists j \in (1,2 \dots v) \tag{19}$$

Let be $f^*_i > \Delta$ a component of f^* which appears in at least one of (19): the existence of such a component is assured in any actual network, given the little Δ value. It follows from the definition of x^*_i that $f^*_i = x^*_i$, and then from (18):

$$c^*_i(f^*) = c_i(f^*_1 \dots f^*_{i-1}, f^*_i - \Delta, f^*_{i+1} \dots f_a) + \frac{\bar{\lambda}\Delta}{\Delta+1} > \frac{\bar{\lambda}\Delta}{\Delta+1}$$

Given a pair $w_n \in W$, a mode $t_h \in \bar{T}_w$ and a path $p_k \in P^{t_h}_{w_n}$ such that $h^*_{p_k} > 0$ and $\delta_{ki} = 1$, the following relation holds:

$$C^*_{p_k}(h^*) = \sum_{j \in L} c^*_j(f^*)\delta_{kj} > \frac{\bar{\lambda}\Delta}{\Delta+1}$$

On the other hand $\lambda^{t_h}_{w_n}(d^*) \in \Lambda$, so that $C^*_{p_k}(h^*) - \lambda^{t_h}_{w_n}(d^*) > 0$. Let be a vector $\begin{bmatrix} h \\ d \end{bmatrix} \in \bar{\Omega}$ such that $h_p = h^*_p \; \forall p \neq p_k$ and $h_{p_k} < h^*_{p_k}$: it follows that $d^t_w = d^{t*}_w$ if $(t,w) \neq (t_h, w_n)$ and $d^{t_h}_{w_n} = d^{t_h *}_{w_n} + h_{p_k} - h^*_{p_k}$. If we consider the expression on the left hand of (8), where $\begin{bmatrix} h \\ d \end{bmatrix}$ is the vector now defined and $\begin{bmatrix} \bar{h} \\ \bar{d} \end{bmatrix} = \begin{bmatrix} h^* \\ d^* \end{bmatrix}$, it becomes:

$$\left[C^*_{p_k}(h^*) - \lambda^{t_h}_{w_n}(d^*)\right] \cdot (h_{p_k} - h^*_{p_k}) < 0$$

so that $\begin{bmatrix} h^* \\ d^* \end{bmatrix}$ is not a solution of (8). The same demonstration can be repeated for all vectors $\begin{bmatrix} h^* \\ d^* \end{bmatrix} \in Z$; so we have that, if one uses the continuous cost functions (16), (17) and (18), the solution of the variational inequality exists and belongs to Ω, thus a network equilibrium exists.

Let be $\bar{f} = A\bar{h} \in \Theta$ the vector of equilibrium flows on links when we use the modified cost functions $c^*(f)$, and let be i a link for which the following relations hold:

$$\bar{f}_i > 0 \qquad \text{if } 0 \le \bar{x}_i \le \Delta$$
$$\bar{f}_i > \bar{x}_i - \Delta \qquad \text{if } \bar{x}_i > \Delta \tag{20}$$

Let consider a new cost function $\hat{c}(f)$ such that $\hat{c}_i(f) = c_i(f)$ for all links but those which satisfy (20). For these we define:

$$\hat{c}_i(f) = c_i(f) + \Delta c_i$$

where

$$\Delta c_i = c_i(\bar{f}_1 \ldots \bar{f}_{i-1}, 0, \bar{f}_{i+1} \ldots \bar{f}_a) + \frac{\bar{\lambda}}{\bar{x}_i + 1} \bar{f}_i - c_i(\bar{f})$$

$$\text{if } \bar{f}_i > 0 \text{ and } 0 \le \bar{x}_i \le \Delta \tag{21}$$

$$\Delta c_i = c_i(\bar{f}_1 \ldots \bar{f}_{i-1}, \bar{x}_i - \Delta, \bar{f}_{i+1} \ldots \bar{f}_a) + \frac{\bar{\lambda}}{\Delta + 1} (\bar{f}_i - \bar{x}_i + \Delta) - c_i(\bar{f})$$

$$\text{if } \bar{x}_i > \Delta \text{ and } \bar{f}_i > \bar{x}_i - \Delta$$

It easy to verify that the vector $C^*(\bar{h})$ due to cost functions $c^*(f)$ coincides with the vector $\hat{C}(\bar{h})$ due to $\hat{c}(f)$, because when $f = \bar{f} = A\bar{h}$ we have $c^*(f) = \hat{c}(f)$. It follows from the definition of equilibrium (12) that, if $\bar{h}$ is an equilibrium pattern for the network with cost function $c^*(f)$, it is an equilibrium pattern also when $\hat{c}(f)$ is the cost function. Thus the values obtained from (21) are the additional costs to be imposed on links in order to assure the network equilibrium.

It is worth noting that the definition of $\bar{\lambda}$ given above is necessary in order to assure that the solution of the variational inequality does not belong to Z in any case; but a rather less $\bar{\lambda}$ value should be used in many actual situations so as to avoid computational difficulties.

In order to illustrate the effect of road pricing on equilibrium of transport systems, let consider the very simple network represented in fig. 1, where a pair of nodes (w_1, w_2) is connected by a link characterized by the cost function $c(f)$. The link flow is equal to the demand between the two nodes; let be $\lambda(d)$ the inverse of demand function, and let be H the link capacity. Fig. 1a shows that no equilibrium solution exists, because for any demand value in the interval $[0,H]$ the link cost $c(f)$ is less than $\lambda(d)$, so that the equilibrium conditions (12) are not verified. Fig. 1a confirms that, only if capacity constraint is a strict inequality, the equilibrium condition (12) are equivalent to variational inequality (13): in fact, if the interval of feasible solution $[0,H]$ were closed, the point $f=H$ would be a solution of the variational inequality, because we would have that, for any $f \in [0,H]$:

$$\left[c(H) - \lambda(H)\right] \cdot (f - H) > 0$$

whereas H is not an equilibrium solution. On the contrary, if the point $f=H$ is not a feasible solution, no point $f^* \in [0,H)$ is a solution of the variational inequality: as a matter of fact an $f \in [0,H)$ always exists, so that:

$$\left[c(f^*) - \lambda(f^*)\right] \cdot (f - f^*) < 0$$

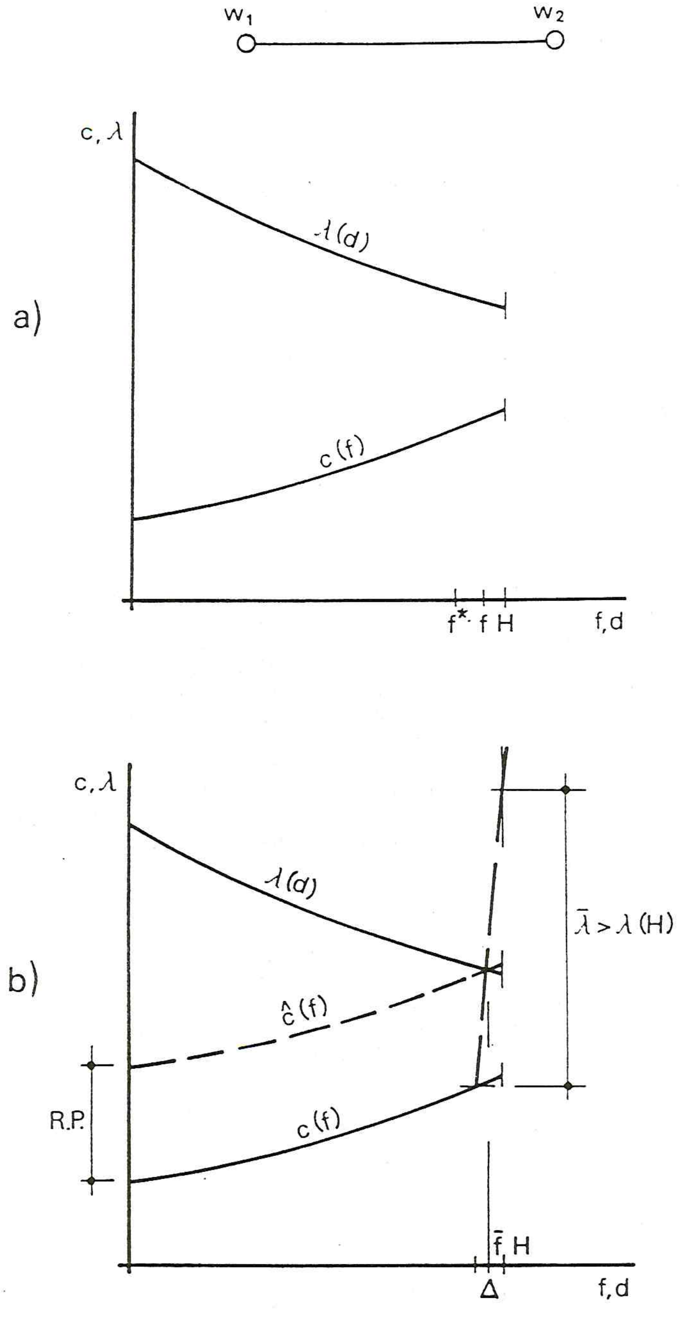


Fig. 1. The use of road pricing to obtain equilibrium

Let be $\bar{\lambda}$ a number greater than the value of the function $\lambda(d)$ in the point $d=H$:

$$\bar{\lambda} > \lambda(H)$$

and let be Δ a positive constant, as little as possible. The cost function is modified in the following way:

$$c^{\bullet}(f) = c(f) \qquad \text{if } f \leq H - \Delta$$

$$c^{\bullet}(f) = c(H-\Delta) + \frac{\bar{\lambda}}{\Delta+1}(f - H + \Delta) \qquad \text{if } f > H - \Delta$$

Fig_1*b* shows that the curves of functions $\lambda(d)$ and $c^{\bullet}(f)$ intersect in a point $\bar{f}$ which is the equilibrium solution. The difference $\Delta c = c^{\bullet}(\bar{f}) - c(\bar{f})$ is the road pricing (*R.P.*), which determines the new cost function on the link, represented by a dashed line in the figure: $\hat{c}(f) = c(f) + \Delta c$.

5. Conclusions

Tha actual use of the results obtained in the previous section needs the transform of the additional cost into money value, i.e. into *road pricing*, and the introduction of a technology, e.g. like that studied for Hong Kong (Catling and Harbord, 1985), which makes it possible to levy a proper tax on each vehicle travelling along each of these links.

However it has to be noted that the application of road pricing calculated by this method on a given network can give rise to an equilibrium that implies modifications of the demand generated by the various origins and of its distribution among the different destinations which could not be acceptable from social and economic points of view. Therefore, in order to make the new equilibrium acceptable, often a modification of the network structure becomes necessary, in which road pricing is integrated with other techniques of traffic control and transport planning (Bell, 1992).

In any case it is necessary that the new network structure is consistent with a management of road pricing which is easy and acceptable by users: this new structure varies following the situations, because it depends on characteristics of urban areas and of demand. Therefore it does not seem possible to transfer the results obtained in a certain town to other urban areas.

REFERENCES

Beckmann M., McGuire C.B. and Winsten C.B. (1956) *Studies in the Economics of Transportation*, Yale University Press, New Haven, CT.

Bell M.G.H. (1992) Future directions in traffic signal control, *Transpn. Res.* 26 A, 303-313.

Catling J. and Harbord B. (1985) Electronic road pricing in Hong Kong: the technology, *Traffic Engng. Control* 26, 608-615.

Dafermos S. (1982) The general multimodal network equilibrium problem with elastic demand, *Networks* 12, 57-72.

Domencich T.A. and McFadden D. (1975) *Urban travel domand*, North Holland, Amsterdam.

Ferrari P. (1994) Road pricing and network equilibrium, (submitted to *Transpn. Res.*).

Glazer A. (1981) Congestion tolls and consumer welfare, *Public Finance* 36, 77-83.

Hills P. (1993) Road congestion pricing: when is it a good policy? A comment, *J. Transport Economics and Policy* 27, 91-99.

Kinderleher D. and Stampacchia G. (1980) *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York, N.Y.

McFadden D. (1975) Conditional logit analysis of qualitative choice behaviour, *Frontiers in Econometrics*, Academic Press, New York, NY, 105-142.

May A.D. (1986) Traffic restraint: a review of the alternatives, *Transpn. Res.* 20 A, 109-121.

Minister of Transport (1963) *Traffic in towns*, Her Majesty's Stationery Office, London.

Morrison S.A. (1986) A survey of road pricing, *Transpn. Res.* 20 A, 87-97.

Quinn D.J. (1992) A review of queue management strategies, *Traffic Engng. Control* 33, 600-605.

Starkie D. (1986) Efficient and politic congestion tolls, *Transpn. Res.* 20 A, 169-173.

Transportation Research Board (1985) *Highway Capacity Manual*, Special Report 209, Washington, D.C.

Treiterer J. and Myers J.A. (1974) The hysteresis phenomenon in traffic flow, *Proceedings of the 6th Int. Symposium on Transport and Traffic Theory*, Reed Pty Ltd, Artamon N.S.W., 13-38.

Waterfield V.H. and Hickmann A.J. (1982) Estimating air pollution from road traffic: a graphical screening method, *TRRL Report* SR 752, Crowthorn.

AN INTEGRATED APPROACH TO VEHICLE AND CREW SCHEDULING

Richard Freling
Tinbergen Institute, Faculty of Economics, Erasmus University Rotterdam, The Netherlands
email: richard@gist.deio.fc.ul.pt

José Paixão
DEIO, Faculdade de Ciências, Universidade de Lisboa, Portugal

TRISTAN II Conference Draft Paper, 22 April 1994

Abstract:
We propose a new formulation and a lower bound procedure for an integrated approach to vehicle and crew scheduling. The vehicle and crew scheduling problem is usually decomposed into subproblems. As a simplification, vehicles are mainly scheduled a priori, because the crew scheduling problem alone is hard enough to solve. After presenting the problem and discussing recent developments in the area, we propose a new mathematical formulation. We focus on obtaining lower bounds by exploiting the network structure of the problem. A column generation approach is suggested, where the column generation subproblem is a two phase approach consisting of an all-pairs shortest path algorithm followed by a depth-first search algorithm. When considering a Lagrangean relaxation of the master problem, the vehicle and crew scheduling aspects are decomposed such that algorithms developed for the vehicle scheduling problem can be applied with a few changes. The paper is concluded with some preliminary computational results for an example, and topics of current and future research.

Keywords: Vehicle Scheduling, Crew Scheduling, Network Models, Set Partitioning, Lagrangean Relaxation, Column Generation.

1. Introduction

In this paper we propose an integrated approach for scheduling vehicles and crews. The *vehicle and crew scheduling problem* (VCSP) is the following: given a set of service requirements or *trips* in the planning horizon, find a feasible *minimum cost* schedule for the vehicles and the crews, while satisfying several crew constraints. The goal is to assign trips to vehicles such that each trip is covered by a vehicle, and to assign work *tasks* to crews such that each vehicle "on the road" is covered by a crew. Most of the existing approaches decompose the problem into subproblems, mainly scheduling vehicles prior to and separately from crews. The reason for this decomposition is the complexity of the problem: the crew scheduling problem in itself is NP-hard (Desrochers and Soumis [1989]), while the vehicle scheduling problem is only polynomial solvable for the most basic case (Lenstra and Rinnooy Kan [1981]). However, recent theoretical developments in the field of optimization, as well as rapid developments in computer technology, in terms of power and speed at much lower cost, inspired us to tackle these problems in an integrated manner. Considerable gains in schedule quality can be obtained if it is possible to exploit the inter-dependency between vehicle and crew schedules. Because crews are relatively inflexible in their movements compared to vehicles, it is often inefficient to schedule vehicles without notice of crews.

Literature about vehicle and crew scheduling concentrates on the applications to urban bus and driver scheduling and airline fleet planning and crew pairing. Several methods to solve both the vehicle scheduling and the crew scheduling problem are discussed in an extended survey by Bodin et al. [1983], a survey based on network models by Carraresi and Gallo [1984], and in a paper showing recent developments in the area of routing and scheduling by Desrosiers et al. [1993]. Developments in mass transit scheduling are also discussed in the proceedings of the last four workshops on computer-aided transit scheduling (Wren [1981], Rousseau [1985], Daduna and Wren [1988] and Desrochers and Rousseau [1992]). Although many mass transit organizations schedule vehicles and crews simultaneously, only a few computerized procedures of this kind are discussed in the literature. Most of these procedures are based on the approach proposed by Ball et al. [1983], whose motivation for integrating vehicle and crew scheduling is the dominance of crew operating costs in mass transit companies.

The approach discussed in this paper is mainly related to the urban bus and driver scheduling problem. The rest of this paper is organised as follows. In the next section the vehicle scheduling problem, the crew scheduling problem and the vehicle and crew scheduling problem are defined and discussed. In section 3, we give a mathematical formulation for the vehicle and crew scheduling problem, combining the network flow structure of the vehicle scheduling problem with a set partitioning formulation for the crew scheduling problem. Algorithms to find a lower bound based on a Lagrangean relaxation approach are presented in section 4. Column generation is used because the number of variables grows very fast with the size of the problem. The Lagrangean relaxation approach allows us to exploit the network structure of the vehicle scheduling problem, while the column generation approach allows us to exploit the network flow structure of the crew scheduling problem. The column generation subproblem, discussed in section 5, is decomposed in a two phase approach based on two acyclic networks. Finally, in section 6 and 7 we show computational results for an example, and we discuss subjects of current and future research, respectively.

2. Problem Description and Literature Review

The input size for vehicle and crew scheduling problems depends on the number of locations where vehicles and/or crews can be substituted. At those locations decisions have to be made for vehicles and crews. The input is determined by the planning horizon, which varies between one day for mass transit scheduling and several days for airline scheduling, and by the physical planning area, which can be one line (line-by-line approach), a cluster or the complete physical network. Figure 1 shows an examnple of a physical network with one depot and three bus line.

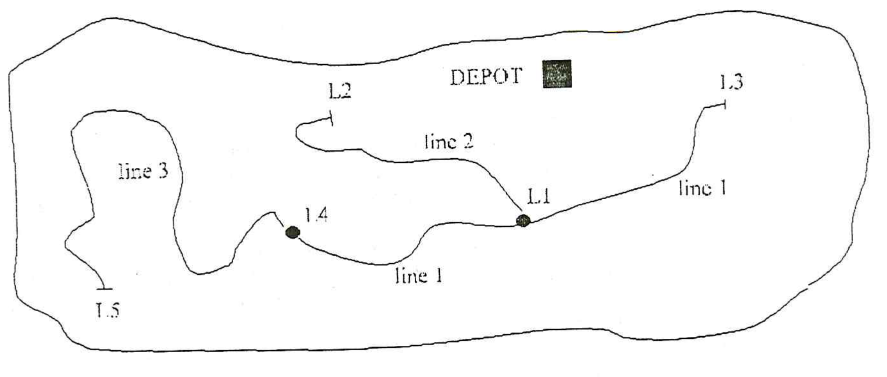


Figure 1 - example of a physical network

Before discussing the integration of vehicle and crew scheduling, we define next the vehicle scheduling problem and the crew scheduling problem. These two problems have a similar structure (assign vehicles to trips or crews to tasks), but crew scheduling generally involves more complicated restrictions (breaks, maximum working time, etc.).

2.1. Vehicle Scheduling

The *single depot vehicle scheduling problem* (SDVSP) is defined as follows: given a depot d and n trips from locations b_i to e_i, with corresponding times bt_i and et_i ($i=1,\ldots,n$), and given the travelling times between all pairs (d,b_i), (b_i,e_i), (e_i,b_j) and (e_i,d), find a feasible *minimum cost* schedule for the vehicles, while all the trips are covered by a vehicle. Each trip has to be entirely serviced by one vehicle and trips are linked with *dead-heading* (*dh*) trips. These are trips *without* serving passengers (pairs (d,b_i), (e_i,b_j) and (e_i,d)), consisting of travel and/or *idle time* (vehicle waiting time). A schedule for a vehicle is composed of vehicle *blocks*, where each block is a departure from the depot, the service of a sequence of pair-wise compatible trips and the return to the depot. The cost function consists of vehicle capital (fixed) and/or operational (variable) costs. An example of a vehicle trip in figure 1 above is a trip on line 2 starting at 7:00 at location *L2* and ending at 7:45 at location *L1*.

We discuss a vehicle scheduling network corresponding to the SDVSP. Let N be the set of trips and E the set of *dh*-trip arcs (e_i,b_j) between *compatible* trips i and j; the nodes s and t represent departure from and arrival to the depot. Two trips i and j are compatible if

$$trav(e_j,b_i) \leq bt_j - et_i,$$

where $trav(e_j,b_i)$ is the dead-heading travel time from locations e_j to b_i. We define the vehicle scheduling network $G=(V,A)$, with nodes $V=N\cup\{s,t\}$ and arcs $A=E\cup(s\times N)\cup(N\times t)$ which is the set of *dh*-trips (inter-trips) and *dh*-trips (s,b_i) and (e_i,t) between each trip and the depot. Figure 2 illustrates this network for four round-trips ($b_i=e_i$) with a duration of one hour each. The network is acyclic and we only need to consider arcs (i,j) with $i<j$, because nodes are ordered by increasing start time of the trips, and we set $s=0$ and $t=n+1$. A path from s to t in the network represents a feasible vehicle block, and a feasible schedule is a set of independent paths (blocks) such that each node (trip) is covered. Fixed costs are defined on each arc leaving the depot, while variable costs are defined on each inter-trip arc.

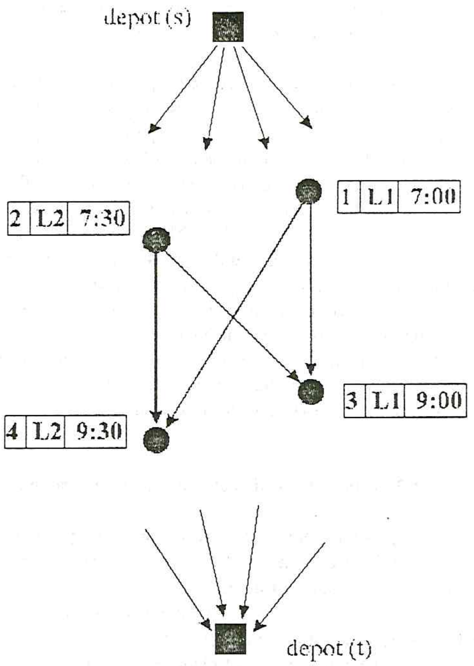


Figure 2 - vehicle scheduling network

Most algorithms for the (polynomial solvable) SDVSP are based on network flow models like minimum cost flow or assignment. An efficient algorithm is the *quasi-assignment* algorithm proposed by Paixão and Branco [1987]. This algorithm uses an improved version of the Hungarian method for assignment problems with complexity $O(mn)$, where m is the number of dead-heading trips (i.e. arcs in the network). Another problem solved by the quasi-assignment algorithm is the SDVSP with a fixed number of vehicles (see Paixão and Branco [1988]). A recent survey on vehicle scheduling can be found in Desrosiers et al. [1993]. Time-space networks are discussed and illustrated for, among others, urban-bus and aircraft fleet assignment. Several heuristic algorithms are presented in the literature for other (NP-hard) variations of the SDVSP. See for example Desrosiers et al. [1993] for the multiple depot case, Freling and Paixão [1994a] for the time constraint case and Ceder [1994] for the case with different types of vehicles. In this paper we consider the SDVSP.

2.2. Crew Scheduling

The goal of the crew scheduling problem (CSP) is to determine a feasible *minimum cost* schedule for the crews such that all work *tasks* are covered by a crew. Each trip and *dh*-trip can be decomposed into tasks by splitting it at relief points corresponding to locations where a change of crew can occur. A task represents the minimum part of service that can be assigned to crews. A *duty* consists of a feasible combinations of *pieces* of work, which in turn are feasible sequences of tasks on *one* vehicle block. The feasibility of duties depends on government rules and the collective agreement between crews and management. We consider *local constraints* for each duty and *global constraints* for a (sub)set of duties. Typically, the local constraints are restrictions on the total working time, the total spread time (duty duration), the length of pieces of work, the length of breaks and the number of pieces of work. Global constraints are for example a maximum average working time, or the case where each daily duty must originate and terminate at a single crew domicile. The cost function implies usually that the total amount of crew wages and/or crew operational costs are minimized. When a duty consists of more than one piece of work, the pieces are separated by breaks and/or unworked periods. As for vehicles also *crew dead-heading*, i.e. transporting a crew *without* serving a vehicle, is possible. In airline scheduling *daily duties* must be grouped in *pairings*, which cover the planning horizon of several days.

When vehicles are scheduled a priori the tasks are determined by splitting the set of vehicle blocks at their relief points. The pieces of work are then determined on each vehicle. In this setting recent successful algorithms are based on set covering or partitioning approaches (Lavoie et al. [1988], Desrochers and Soumis [1989], Paixão [1990] and Hoffman and Padberg [1993]). The last authors present a branch-and-cut approach which enables them to solve problems up to 825 rows (flight legs) and up to 1.05 million variables with duality gap less than 1%. A unified approach to crew scheduling based on set partitioning and network models is proposed by Carraresi et al. [1994]. They use Lagrangean relaxation and column generation to solve problems up to 1552 rows (flight legs) and 2156.4 million variables with duality gap less than 5%. A greedy heuristic is used to obtain an upper bound. Current research is concentrating on closing the gap further by improving the upper bound procedure using logic programming. Another unified approach to crew scheduling based on a multi-commodity flow formulation is presented by Desrosiers et al. [1993]. Dantzig-Wolfe decomposition applied to this formulation leads to a similar approach as column generation applied to the set partitioning formulation. They report that recently they were able to solve airline crew pairing problems with up to 4000 flight legs using time-slice decomposition.

2.3. Integration of Vehicle and Crew Scheduling

In general, vehicle and crew scheduling problems interact with each other: the specification of vehicle schedules will set certain constraints on the crew schedules and vice versa. To get some more insight in the vehicle and crew scheduling problem, we say that crews are *tight* to vehicles if the specification of crew schedules is sensitive to the way vehicles are scheduled. The tightness of crews to vehicles depends mainly on the following integration components:

- a restricted number of changes of vehicle per duty;
- no crew dead-heading or extra crew dead-heading travel time;
- startup time on a vehicle;
- continuous attendance when a vehicle is waiting;
- minimum duration of a piece of work.

In section 5.3 we discuss the impact of these components on the suggested approach. The extra-urban bus driver scheduling problem (see Tosini and Vercellis [1988]) is an example of the case where crew dead-heading is not allowed. If vehicles are scheduled without notice of crews, it may be that no feasible crew schedule exists at all, because a vehicle can be "on the road" to long to be serviced by a crew. Therefore, the vehicle should pass by a crew domicile once in a while to substitute the crew. Figure 3 below illustrates this case with a set of trips on one vehicle block, marked by their locations. Suppose that crew reliefs can only occur at location *L1*. Then, if the duration of the four trips away from location *L1* is more than the maximum length allowed for a piece of work, the problem turns infeasible.

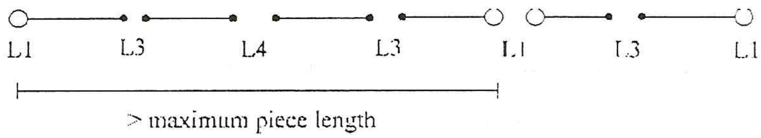


Figure 3 - example infeasible vehicle block with 6 trips

Each of the mentioned components appear frequently in practice and influence the efficiency of scheduling vehicles and crews together. This leads to the conclusion that it often is inefficient to schedule vehicles without notice of crew schedules, because the scheduling of vehicles is much more flexible compared to the scheduling of crews. Note that in the unpractical case where none of the components occur, there is no need to integrate vehicle and crew scheduling because crews can move through the network independently of vehicles. In the remaining of this section, we will discuss the literature on a simultaneous approach to mass transit scheduling. No literature is known to the authors on simultaneous airline scheduling. A common argument for the sequential airline scheduling approach is that vehicle costs dominate crew costs. But saving a small percentage of crew costs using the integrated approach can produce a large profit, because airline crew operating costs are generally very high. Besides that, crews are often tight to vehicles because of a startup time and the expenses of crew dead-heading (e.g. overnight costs).

The traditional sequential strategy is strongly criticized by Bodin et al. [1983], who state that in the public transport case the crew costs dominate vehicle operating costs and in some cases reach as high as 80% of total operating costs. This argument motivated Ball et al. [1983] to propose a heuristic procedure to schedule vehicles and crews simultaneously. Their algorithm involves forming a scheduling graph, which consists of vertices characterized by parts of trips called *d-trips* that have to be executed by one vehicle and crew, and two vertices s and t representing the depot. Several types of arcs are of two kinds, those which indicate that a crew and vehicle proceed from one d-trip to another and those which indicate that only the crew proceeds from one d-trip to another. Because the partitioning model based on this graph is too complex, they use a decomposition approach emphasizing the crew scheduling problem. The solution is decomposed into three components: a piece construction component, a piece improvement component and a duty generation component. All three components are solved using matching algorithms. The piece construction routine generates a set of pieces whose time duration is less than some constant *T*, while vehicle schedules are generated simultaneously by deleting the "crew-only" arcs. The duty generation is done by fixing crew-only arcs, where in a first step pairs of short pieces are combined into partial 3-piece duties, and in a second step pairs of these duties and longer pieces are combined into 2 and 3-piece duties. They succeeded in obtaining a "reasonable" solution for the VCSP on the whole physical network, while placing no restrictions on interlining.

Other papers appeared in the literature on a (quasi-)simultaneous approach to mass transit scheduling. They are of two types: vehicle schedules are determined with notice of crews and then crews are scheduled afterwards, or vehicles are scheduled during the crew scheduling process. Approaches of the first type are proposed by Scott [1985] for an extension to the HASTUS crew scheduling model and by Darby-Dowman et al. [1988] as part of a decision support system. Note that the matching approach from Ball et al. [1983] described above is of the second type. Other approaches of the second type are proposed by Tosini and Vercellis [1988], Falkner and Ryan [1992], and by Patrikalakis and Xerocostas [1992].

The aim of our research is the development of a fully integrated approach to vehicle and crew scheduling. Next, we propose a mathematical formulation designed in this context.

3. Mathematical Formulation for the VCSP

The key to formulating the VCSP is the way crew tasks are defined without knowing the vehicle schedules. We consider two types of tasks, those on trips of which we can be sure that they have to be serviced by a crew, and those on vehicle dead-head trips that only need to be covered by a crew if a vehicle is traversing this trip. The formulation is in fact an integration of the quasi-assignment formulation for vehicle scheduling based on network $G=(V,A)$ presented in section 2.1, and the set partitioning formulation for crew scheduling.

Related to the formulation we consider the following definitions:

- c_{ij} is the vehicle cost of arc $(i,j)\in A$;
- d_k the cost of duty $k\in K$;
- I_n is the set of crew tasks on all vehicle trips (trip tasks);
- $I(i,j)$ is the set of crew tasks on vehicle dead-head trip $(i,j)\in A$ (*dh*-trip tasks);
- $I = I_n \cup (\bigcup_{(i,j)\in A} I(i,j))$ is the set of all crew tasks;
- $K(i)$ the set of duties covering task $i\in I$;
- decision variable $y_{ij}=1$ if a vehicle covers trip j directly after trip i, $y_{ij}=0$ otherwise;
- decision variable $x_k=1$ if duty k is selected, $x_k=0$ otherwise.

The vehicle capital cost is included in arcs emanating from s, while the vehicle operational costs are included in the remaining arcs. The VCSP is then formulated as follows (model QASP):

$$\min \sum_{(i,j)\in A} c_{ij}\, y_{ij} + \sum_{k\in K} d_k\, x_k$$

$$\sum_{\{j:(i,j)\in A\}} y_{ij} = 1 \qquad \forall\, i\in N \tag{i}$$

$$\sum_{\{i:(i,j)\in A\}} y_{ij} = 1 \qquad \forall\, j\in N \tag{ii}$$

$$\sum_{j\in N} y_{n+1,j} \leq v \tag{iii}$$

$$\sum_{k\in K(i)} x_k = 1 \qquad \forall\, i\in I_n \tag{iv}$$

$$\sum_{k\in K(q)} x_k = y_{ij} \qquad \forall q\in I(i,j),\ \forall\ (i,j)\in A \tag{v}$$

$$y_{ij},\ x_k \in\{0,1\} \qquad \forall\ (i,j)\in A, \forall\ k\in K \tag{vi}$$

The objective is to minimize vehicle and crew costs. The first three constraints correspond to the quasi-assignment formulation of the SDVSP. Constraints (i) and (ii) are typical vehicle assignment constraints, assuring that each trip will be assigned to a vehicle. Constraint (iii) includes the possibility to restrict the number of vehicles with a fixed parameter v. Constraints (iv) and (v) are set partitioning constraints, where constraints

(iv) guarantee that exactly one duty k of the set $K(i)$ covers each task on a trip, while constraints (v) ensure that each *dh*-trip task corresponding to a *dh*-trip in the solution is covered by a duty. Global crew schedule constraints $x \in X$ can be added if necessary, where X is the set of global constraints (see Carraresi et al. [1994]).

4. Lower Bounds

We propose a procedure to obtain lower bounds with the use of **optimization** techniques exploiting the special time-space network structure of the VCSP. We discuss a **column generation** algorithms based on a Lagrangean relaxation for model QASP presented in the previous section. This model contains usually a huge number of variables corresponding to feasible duties. Clearly, column generation is necessary to deal with these variables. A sophisticated approach to implicitly consider all the feasible duties is an implicit column generation approach proposed by Desrochers and Soumis [1989] for the urban transit case and by Lavoie et al. [1988] for the airline case. Starting with a basic set of duties (columns) one generates iteratively new columns by solving a subproblem that determines negative reduced cost duties. This method makes the set covering or partitioning approach computational more attractive because the set of duties in a solution is relatively small. Desrosiers et al. [1993] show that column generation is a very powerful tool for many NP-hard routing and scheduling problems. They report for example that for a crew pairing problem with 282 flight legs about 20000 pairings were generated, although the estimated total number of feasible pairings was $190 * 10^{12}$. Another advantage of this approach is that most real-world situations can be adequately modeled within its framework. More specifically, since columns are implicitly assumed to represent feasible duties, virtually any criteria of duty feasibility can be enforced by permitting only feasible columns to enter the formulation. In this section we apply the implicit column generation algorithm to the VSCP, while we discuss the network construction and solution approach for the column generation subproblem in section 5.

4.1. Column Generation

The idea of column generation is to consider only subsets of the complete set of columns, i.e. in our case the set of duties K. Implicitly we take account of the entire set of columns, because in each iteration we look for new columns among all possible columns. A relaxation with a temporarily subset of columns in an iteration is the *master problem*, while the generation of a new subset of columns is the *subproblem*. This subproblem returns a (sub)set of nonpositive reduced cost columns to the master problem. The column generation algorithm converges when the subproblem does not return any nonpositive reduced cost columns. The current set of columns is then optimal for the relaxation. Table 1 illustrates the column generation algorithm.

step 1. Determine an initial set of columns;

step 2. **Master Problem:** Determine a new dual solution based on the current set of columns and calculate the corresponding reduced costs;

step 3. **Subproblem:** Use the reduced costs to generate a new set of columns. If no nonpositive reduced cost columns have been found, then stop. Otherwise return to step 2.

Table 1 -column generation algorithm

The column generation approach to the VCSP is similar to that of the crew scheduling problem. The column generation master problem for crew scheduling is generally a linear programming (LP) relaxation, which is solved with the simplex method (see Desrochers and Soumis [1989]). Starting from an initial basic set of columns (K_1), one iteratively solves the lp relaxation and uses the dual variables (simplex multipliers) to determine negative reduced cost columns to add to the initial set of columns. We will refer to the lower bound obtained with the LP relaxation as the master problem as V_{LP} and to the corresponding algorithm as CGLP.

4.2. Lagrangean Relaxation

Instead of the LP relaxation, we can use a Lagrangean relaxation to take more advantage of the special structure of the VCSP. Cattryse et al. [1993] apply this approach to the discrete lotsizing problem. Their algorithm is similar to CGLP but with simplex replaced by dual ascent/subgradient optimization, while the dual variables are replaced by Lagrangean multipliers. The second algorithm for the VCSP is related to an algorithm proposed by Carraresi et al. [1994], who apply the Lagrangean approach to the crew scheduling problem. A complete new subset of nonpositive reduced cost columns is generated in each iteration of their algorithm. This approach is justified because the Lagrangean relaxation only considers variables (duties) with nonpositive reduced cost.

Consider model QASP presented in section 3. With $\mathscr{F}(QA)$ we denote the network flow (quasi-assignment) constraints, (i), (ii), (iii) and the integrality condition of the y variables. The Lagrangean relaxation is obtained after relaxing the set partitioning constraints (iv) and (v) by including them in the objective function:

$$\mathrm{L}(\lambda,\mu) = \min \sum_{k\in K} d_k\, x_k + \sum_{i,j\in A} c_{ij}\, y_{ij} + \sum_{i\in I_n} \lambda_i \,(1 - \sum_{k\in K(i)} x_k) + \sum_{i,j\in A} \; \sum_{q\in I(i,j)} \mu^q_{ij}\, (y_{ij} - \sum_{k\in K(q)} x_k)$$

subject to $\mathscr{F}(QA)$ and $x_k \in \{0,1\}\ \forall k\in K$;

Let $I_n(k)$ and $A(k)$ be the set of trip tasks in I_n^- and the set of dh-trip arcs in A covered by duty $k\in K$, respectively. Then we define:

- The reduced cost with respect to vehicle arc $(i,j)\in A$: $\bar{c}_{ij} = c_{ij} + \sum_{q\in I(i,j)} \mu^q_{ij}$;
- The reduced cost with respect to duty $k\in K$: $\bar{d}_k = d_k - \sum_{i\in I_n(k)} \lambda_i - \sum_{(i,j)\in A(k)} \; \sum_{q\in I(i,j)} \mu^q_{ij}$;

and $\mathrm{L}(\lambda,\mu)$ can be written as:

$$\mathrm{L}(\lambda,\mu) = \sum_{i\in I_n} \lambda_i + \min\{ \sum_{i,j\in A} \bar{c}_{ij}\, y_{ij} \mid \mathscr{F}(QA)\} + \min\{ \sum_{k\in K} \bar{d}_k\, x_k \mid x_k \in \{0,1\}\} \tag{4.1}$$

Let $y^*(\lambda,\mu)$ and $x^*(\lambda,\mu)$ be an optimal solution to $\mathrm{L}(\lambda,\mu)$ for given λ and μ. The Lagrangean function $\mathrm{L}(\lambda,\mu)$ can be evaluated by solving the SDVSP to yield $y^*(\lambda,\mu)$ and by verifying $|K|$ inequalities $\bar{d}_k \leq 0$ to yield $x^*(\lambda,\mu)$. If global constraints of the form $x\in X$ are included, they can often be handled by manipulating the verification of the reduced cost (ordered lists, etc.). If this is not possible, we can include them in the Lagrangean relaxation. A lower bound V_{LAG} on the optimal solution can be obtained by maximizing the Lagrangean function:

$$V_{LAG} = \max\ \mathrm{L}(\lambda,\mu) \tag{4.2}$$

This *Lagrangean dual problem* is a nondifferentiable optimization problem, which is generally solved by subgradient optimization when it occurs in the context of Lagrangean relaxation. A powerful alternative for subgradient optimization is a bundle trust method developed by Schramm [1989]. We briefly discuss the idea behind this method. For more reading on this subject we refer to Hiriart-Urruty and Lemaréchal [1993] for general theory on bundle methods, and to Schramm [1989] for the bundle trust method in particular. Generally, nondifferentiable optimization algorithms iterate from a start solution to some solution "close" to or at the optimum. The update of the variables in each iteration depends on a search direction, which is determined by one or more *subgradients*, and a step size along the direction. Bundle methods try to improve the information at a nondifferentiable iteration point by collecting subgradients of previous iteration points in a bundle. It can

be shown that this bundle is an approximation of the subdifferential at the current iteration point. A new search direction is calculated, using a quadratic model, based on the information from the bundle. This quadratic model is a cutting plane model with an additional trust region term for the purpose of stabilization (i.e. to avoid *zigzags*). The resulting search direction is a convex combination of the subgradients in the bundle. Recently, Frangioni [1993] has developed a specialized quadratic programming approach for bundle trust algorithms arising in Lagrangean relaxation applications of integer programming problems.
Other characteristics of the bundle trust method are:

- the step size is computed by a heuristic strategy;
- the size of the bundle is controlled because the size of the quadratic model depends on it;
- a stopping criterion is available which allows to prove approximate optimality.

Because the evaluation of the Lagrangean function (4.1) will always produce integer variables, the *integrality property* for Lagrangean relaxation holds (see Geoffrion [1974]). Therefore, the lower bounds obtained with the Lagrangean relaxation (V_{LAG}) can not be better than the lower bounds obtained with the Linear relaxation (V_{LP}). We refer to the resulting column generation algorithm with a Lagrangean relaxation in the master problem as CGLR. Both the column generation algorithm and the bundle trust algorithm for the Lagrangean dual are iterative methods. When the algorithm terminates, the set of columns are optimal for the Lagrangean dual problem (4.2). The algorithm returns besides a lower bound an approximately optimal vehicle schedule (with respect to the VCSP) and a "good" set of duties.

5. Column Generation Subproblem

The column generation subproblem for crew scheduling aims to generate a (sub)set of nonpositive reduced cost duties. For our case, the subproblem is decomposed in a piece generation and a duty generation process. The idea is to construct two networks, the first once and for all with paths corresponding to reduced cost pieces, the second based on previously generated pieces with paths corresponding to reduced cost duties. In this section we discuss the construction of the subproblem. We refer to Freling and Paixão [1994b] for a general discussion on solution methods for the column generation subproblem for crew scheduling.

In the following we assume that the cost of each duty k is a separable function of tasks, that is:

$$d_k = \sum_{i \in I_n(k)} \gamma(i) + \sum_{(i,j) \in A(k)} \sum_{q \in I(i,j)} \gamma(q) \ ,$$

where $\gamma(i)$ is any real function of $i \in I$ (e.g. the duration of task i).

5.1. Network Associated with Piece Generation

The first crew network is an extension of the vehicle network G with tasks defined on trips and *dh*-trips. We define network $G^p=(N^p,A^p)$, where nodes correspond to relief points including a source s and a sink t corresponding to the depot. Arcs $(i,j) \in A^p$ correspond to tasks between relief points i and j, to sign-in from the source to each relief point and to sign-out from each relief point to the sink. Arcs with a "long" time duration are not included because they correspond to tasks to and from the depot[1]. Figure 4 illustrates the network for an example with two tasks on each trip and one task on each *dh*-trip (compare this figure with figure 2 in section 2.1). Each path between two nodes on network G^p corresponds to a feasible piece of work if it satisfies the time window [*min_piecetime*, *max_piecetime*], which are the minimum and maximum durations of a piece.

[1] *It is important to note that some dh-trip arcs in vehicle network G can be so-called "long arcs", where in practice a vehicle passes by the depot after servicing a trip and waits there until it has to proceed with the next trip.*

A piece p can be characterized by (b_p,e_p,bt_p,et_p), its begin location, end location, begin time and end time. The time window constraint for a piece of work p is then stated as:

$$(c_1) \qquad min\_piecetime \leq et_p - bt_p \leq max\_piecetime;$$

Associate the length $\delta_i = \gamma(i) - \lambda_i$ with tasks $i \in I_n$ and $\delta_q = \gamma(q) - \mu^q_{ij}$ with tasks $q \in I(i,j)$ for $(i,j) \in A$. Now we define the costs for the network such that paths correspond to reduced cost pieces. Let $P(u,v)$ be a path from u to v on network G^p, then the reduced cost of this path (piece p) is

$$\beta_p = \sum_{i \in P(u,v)} \delta_i$$

(the sum of the reduced costs of the tasks on the path). Thus, a path on network G^p is a feasible piece of work p with reduced cost β_p if the path satisfies condition (c_1).

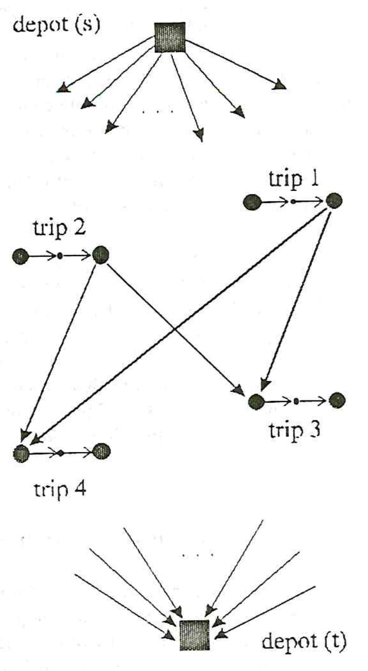


Figure 4 - piece generation network

Note that the network is acyclic because it is impossible to go back in time. Therefore, we can assume that only arcs (i,j) exist with i<j. Pieces with reduced cost smaller or equal to a parameter Δ are generated by solving shortest path problems between each pair of nodes in network G^p that satisfy the constraints on the piece duration. This parameter is increased during the column generation algorithm to control the number of pieces considered for the generation of duties. The problem to find a subset of paths (pieces) with nonpositive reduced cost in network G^p is thus an all-pairs shortest path problem. Condition (c_1) is always satisfied because only paths $P(u,v)$ are generated for nodes u and v satisfying the condition. Because the network is acyclic we can use a reaching algorithm to find the shortest path from node u to v in $O(m)$ time, where m is the number of arcs in network G^p (see Ahuja et al. [1993]). The total complexity of the piece generation is then $O(mn)$.

5.2. Network Associated with Duty Generation

Consider a set of pieces W determined in the previous network and ordered by increasing start time. Associated to crew duties define the network $G^d = (N^d, A^d)$, where nodes N^d correspond to pieces of work (W) and a source, while an arc $(p,q) \in A^d$ corresponds to compatible pieces of work p and q. Two pieces of work are compatible if they can be assigned, in that order, to the same crew duty. For example pair $(e_p,b_q) \in A^d$ is compatible if

$$crewtrav(e_p,b_q) + min\_break \leq et_p - bt_q,$$

where $crewtrav(e_p,b_q)$ is the crew dead-heading travel time from e_p to b_q and min_break is the minimum break time for a crew. Additional arcs from the source to each node indicate the start of a duty. The "inter-piece" arcs correspond to crew breaks, crew dead-heading and crew waiting time. Figure 5 illustrates this network for an example with 6 pieces. A path from the source to any other node on G^d corresponds to a feasible duty if local crew constraints are satisfied. We assume that these constraints are separable functions of pieces of work, such that the constraint value of a piece corresponds to a resource consumption. For example a resource can correspond to the spread time of a duty, the working time of a duty or the number of pieces in a duty. Each arc $(i,j) \in A^d$ consumes an amount $d^r_{ij} \geq 0$ of each resource $r = 1,\dots R$, where R is the number of resources. The consumption of resource r along path P is then $d^r(P) = \Sigma_{(i,j) \in P}\ d^r_{ij}$, and the constraints for a path P are:

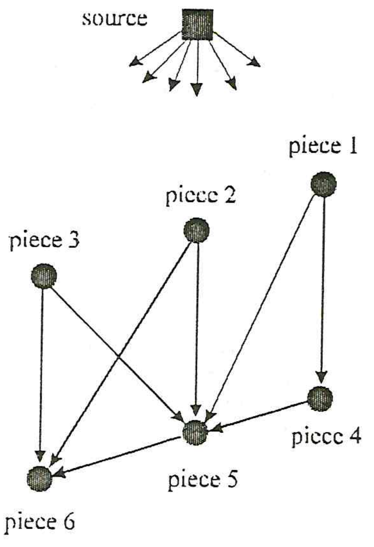


Figure 5 - duty generation network

$$(c_2) \qquad min_r \leq d^r(P) \leq max_r \qquad \text{for } r=1,\ldots,R;$$

where min_r and max_r are the minimum and maximum consumption allowed for each resource, respectively. Furthermore, associate the length β_p (i.e. the reduced cost of piece p) with each node $p \in N^d$. The reduced cost $\bar{d}_k$ of duty k is defined as the sum of the costs of the nodes of a feasible path. Let P_r be a path from the source s to node r on network G^d , then the reduced cost of this path (corresponding to a duty k) is

$$\bar{d}_k = \sum_{p \in P_r} \beta_p$$

This network is also acyclic. The problem to find a subset of duties with nonpositive reduced cost is a resource constrained shortest path problem (see Desrochers [1988], Carraresi et al. [1994]). However, when constraints are tight it can be shown that a simple depth-first search algorithm has worst-case complexity $O(\alpha^2(m-n)M^{\alpha-2})$, where α is the maximum number of pieces allowed in a duty, m and n are the number of arcs and nodes in network G^d, and M is the maximum *outdegree* of all nodes $p \in N^d$ except the source (see Freling and Paixão [1994b]). This complexity is polynomial if the number of pieces is restricted. In urban driver scheduling the maximum number of pieces is generally restricted by 3 or 4.

5.3. Consequences for the integration components

Let us see how networks G^v and G^d relate to the components influencing the inter-dependency between vehicle and crew schedules that we discussed in section 2.3. We first consider continuous attendance. When a *dh*-trip arc in network G^v is not a "long arc", but long enough to include a crew break (because the vehicle is waiting) and continuous attendance is not necessary, this arc is deleted and replaced by two nodes (relief points) and four corresponding arcs. This is illustrated in figure 6.

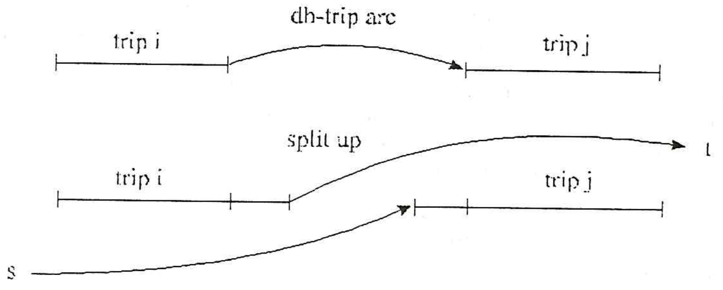


Figure 6 - split up for crew break

The remaining *tightness* components are incorporated in arcs or paths of networks G^v and G^d:
For a restricted number of changes of vehicle the number of nodes (pieces) in a path on network G^d is restricted to α (the maximum number of pieces). The transfer from one piece to another is not

always a change of vehicle, but by restricting the number of pieces, the number of changes of vehicle are restricted as well;

- When crew dead-heading is not alowed pieces in network G^p begin and end at a crew domicile, while network G^d contains only arcs between equivalent locations. Hence, if piece p begins at b_p and ends at e_p then b_p and e_p must be crew domiciles, while if a duty contains pieces p and q, then e_p must be equal to b_q.
- Extra crew dead-heading travel time is reflected on the determination of feasible arcs in network G^d;
- With a startup time on a vehicle (or changeover time) the duration of each piece in network G^p is extended;

6. Computational Results for an Example

For the time being, we show some computational results obtained with preliminary versions of CGLP, and we compare results for a subgradient algorithm and a bundle trust algorithm applied to the Lagrangean relaxation (without column generation) of model QASP. The tests are run for a small example with 7 trips. In each column generation iteration we add the complete set of nonpositive reduced cost columns.

We obtained the following results:

include all duties	column generation
cost 8270, fractional solution	cost 8270, integer solution
250 duties	25 duties, 6 subproblems
346 simplex iterations	71 simplex iterations

Table 2: column generation

The left side of this table shows results for a direct application of the linear programming relaxation with enumeration of all the duties, while the right side shows results obtained with CGLP starting from a basic (greedy) heuristic solution with 15 duties and costs 9285. Note that with the column generation approach applied to this example, the final set of duties contains about 10% of the total number of feasible duties and we need about 20% of the simplex iterations compared to solving the linear programming relaxation directly.

The Lagrangean relaxation without column generation (this is the Master problem in CGLR with the complete set of duties), is tested with an implementation of the subgradient algorithm and an implementation of the bundle trust algorithm, the BT code, developed by Schramm [1989]. Table 3 below shows results for the same example as described above, using an upper bound of 8270 and zero as starting value for the multipliers.

Though the bundle trust algorithm needs more CPU time per iteration, it is obvious that the BT algorithm outperforms the subgradient algorithm for this example. An advantage of the subgradient algorithm is its simplicity of implementation.

subgradient method	bundle trust method
lower bound 8265	lower bound 8270
# iterations 1000	# iterations 99
CPU 21 minutes	CPU 5 minutes

Table 3: Subgradient versus bundle trust

7. Conclusions

We presented a new mathematical formulation for the vehicle and crew scheduling problem based on quasi-assignment and set partitioning constraints. We also suggested an advanced decomposition approach to obtain a lower bound for the vehicle and crew scheduling problem. This approach consists of column generation (i.e. decomposition of columns) with a decomposition of the subproblem by generating pieces of work and duties separately. Furthermore, Lagrangean relaxation decomposes the vehicle and crew part by relaxing the set partitioning constraints.

Future work is dedicated to further decomposition of the rows corresponding to crew tasks. Carraresi et al. [1994] do something similar for the crew scheduling problem. Their approach to obtain upper bounds can be applied to the VCSP. For example, when algorithm CGLR terminates, we can use the corresponding vehicle schedule as a base for determining a feasible crew schedule using a set partitioning formulation.

We are currently testing the approach presented in this paper to a data set of the public transport company of Rotterdam. As a basic start solution we approach vehicle and crew scheduling separately. At the same time, this also allows us to compare these results with the results of the integrated approach.

Though we concentrated so far on the urban bus and driver scheduling problem, we are also planning to apply this type of approach to the airline fleet planning and crew pairing problem.

References

Ahuja, R.K.; Magnanti, T.L. and Orlin, J.B. (1993), *Network Flows, Theory, Algorithms, and Applications*, Prentice Hall, (New Jersey)

Ball, M.; Bodin, L.;Dial, R. (1983), "A Matching Based Heuristic for Scheduling Mass Transit Crews and.Vehicles," *Transportation Science*, 17, p 4-31

Bodin, L.; Golden, B.; Assad, A. and Ball, M. (1983), "Routing and Scheduling of Vehicles and Crews: The State of the Art," *Computers and Operations Research*, 10 (2), p 63-211

Carraresi, P. and Gallo, G. (1984), "Network Models for Vehicle and Crew Scheduling," *EJOR*, 16, p 139-151

Carraresi, P.; Girardi, L and Nonato, M. (1994), "Network Models, Lagrangean Relaxation and Subgradients Bundle Approach in Crew Scheduling Problems," to appear in the proceedings of the 6th International Workshop on Computer-Aided Transit Scheduling (Springer Verlag)

Cattrysse, D.; Salomon, M.; Kuik, R. and Van Wassenhove, L.N. (1993), "A Dual Ascent and Column Generation Heuristic for the Discrete Lotsizing and Scheduling Problem with Setup Times," *Management Science*, Vol. 39, No. 4, p 477 - 486

Ceder, A. (1994), "Vehicle Scheduling with Different Types of Vehicles," to appear in the proceedings of the 6th International Workshop on Computer-Aided Transit Scheduling (Springer Verlag)

Daduna, J.R. and Wren, A. (ed.) (1988), *Computer-aided Transit Scheduling: Proceedings of 4th International Workshop*, Springer Verlag, Berlin

Darby-Dowman, K; Jachnik, J.K.; Lewis, R.L. and Mitra, G. (1988), "Integrated Decision Support Systems for Urban Transport Scheduling: Discussion of Implementation and Experience," in: J.R. Daduna and A. Wren (ed.), *Computer-aided Transit Scheduling: Proceedings of 4th International Workshop*, (Springer Verlag, Berlin), p 226-239

Desrochers (1988), "Shortest path problems with resource constraints," Report G-88-27 GERAD, Montréal

Desrochers, M. and J.M. Rousseau (ed.) (1992) *Computer-aided Transit Scheduling: Proceedings of 5th International Workshop*, Springer Verlag, Berlin

Desrochers, M. and Soumis, F. (1989), "A column generation approach to the urban transit crew scheduling problem," *Transportation Science*, Vol.23, No.1, p 1-13,

Desrosiers, J.; Dumas, Y.; Solomon, M.M. and Soumis, F. (1993), "Time Constrained Routing and Scheduling", Report G-92-42 GERAD, Montréal

Falkner, J.C. and Ryan, D.M. (1992), "EXPRESS: Set Partitioning for Bus Crew Scheduling in Christchurch," in: M. Desrochers and J.M. Rousseau (ed.), *Computer-aided Transit Scheduling: Proceedings of 5th International Workshop*, Springer Verlag, Berlin, p 359-378

Frangioni, A. (1993), "An Efficient Active-Set Algorithm for Finding Descent Directions in Nonsmooth Optimization Algorithms", Working Paper, Department of Computer Science, University of Pisa, Italy

Freling, R. and Paixão, J. (1994a), "Vehicle Scheduling with Time Constraint," to appear in the proceedings of the 6th International Workshop on Computer-Aided Transit Scheduling (Springer Verlag)

Freling, R. and Paixão, J. (1994b), "The Column Generation Subproblem for Crew Scheduling," in preparation

Geoffrion, A.M. (1974), "Lagrangean Relaxation for Integer Programming," *Mathematical Programming Study*, 2, p 82-114

Hiriart-Urruty, J.B. and Lemarechal, C. (1993), *Convex Analysis and Minimization Algorithms*, Vol. I and II, Springer Verlag, Berlin

Hoffman, K.L. and Padberg, M. (1993), "Solving Airline Crew Scheduling Problems by Branch-and-Cut," *Management Science*, 39 (6), p 657-682

Lavoie, S.; Minoux, M. and Odier, E. (1988), "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation," *EJOR*, 35, p 45-58

Lenstra, J.K. and Rinnooy Kan, A.H.G (1981), "Complexity of Vehicle Routing and Scheduling Problems," *Networks*, 11, p 221-227

Paixão, J. and Branco, I.M. (1987), "A Quasi-Assignment Algorithm for Bus Scheduling," *Networks*, vol. 17, p 249 - 269

Paixão, J. and Branco, I.M. (1988) "Bus Scheduling with a Fixed Number of Vehicles," in: J.R. Daduna and A. Wren (ed.), *Computer-aided Transit Scheduling: Proceedings of 4th International Workshop*, (Springer Verlag, Berlin), p 28-40

Paixão, J. (1990), "Transit Crew Scheduling on a Personal Workstation," *Operational Research '90, Selected Papers from the Twelfth IFORS International Conference on Operational Research*, Athens, Greece, p 421 - 432

Patrikalakis, I. and Xerocostas, D. (1990), "A new Decomposition Scheme of the Urban Public Transport Scheduling Problem," in: M. Desrochers and J.M. Rousseau (ed.), *Computer-aided Transit Scheduling: Proceedings of 5th International Workshop*, (Springer Verlag, Berlin), p 407-425

Rousseau, J.M. (ed.) (1985), *Computer Scheduling of Public Transport 2*, North Holland, Amsterdam

Schramm H. (1989), *Eine Kombination von Bundle- und Trust-Region-Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*. Bayreuther Mathematische Schriften, Heft 30, Bayreuth

Scott, D. (1985) "A Large Linear Programming Approach to the Public Transport Scheduling and Cost Model," in: J.M. Rousseau (ed.), *Computer Scheduling of Public Transport 2*, (North Holland, Amsterdam), p 473-491

Tosini, E and Vercellis, C. (1988), "An Interactive System for Extra-Urban Vehicle and Crew Scheduling Problems," in: J.R. Daduna and A. Wren (ed.), *Computer-aided Transit Scheduling: Proceedings of 4th International Workshop*, (Springer Verlag, Berlin), p 41-53

Wren, A. (ed.) (1981), *Computer Scheduling of Public Transport*, North Holland, Amsterdam

A Generalized Labelling Algorithm for the Dynamic Assignment Problem

Warren B. Powell
and
Derek H. Gittoes

Department of Civil Engineering
and Operations Research
Princeton University
Princeton, NJ 08544

May 27, 1994

Abstract

We develop a solution strategy for the dynamic assignment problem, an important class of routing and scheduling problems with many applications in transportation and logistics. A label setting rolling horizon algorithm (LASER) was developed for the solution of the dynamic assignment problem. An extensive numerical analysis on a variety of randomly generated and real-world problems was conducted in order to evaluate the algorithm. The breadth-first LASER algorithm was compared with a depth-first tour construction algorithm and an LP-optimal Dantzig-Wolfe column generation procedure. The LASER algorithm proved to be effective when compared to the other solution procedures from a solution quality and computational efficiency standpoint for most problems. The labelling algorithm however, did not utilize vehicles efficiently which resulted in solution quality deterioration when the problems were constrained with respect to available fleet capacity.

1 Dynamic Assignment Problem Overview

Dynamic routing and scheduling problems constitute a significant class of problems in the fields of transportation and logistics. Numerous real-world applications and intrinsic computational complexity are motivating factors for studying these problems. In many industries transportation and logistics costs comprise a large percentage of overall operating expenses, and have been highlighted as areas where new technologies can be exploited in order to increase overall productivity [3]. This paper develops and evaluates a new solution strategy for a particular subclass of dynamic routing and scheduling problems, namely the dynamic assignment problem (DAP). The subsequent sections describe the DAP and its applications, and present an overview and a list of contributions for the paper.

Within the motor carrier industry are several important routing and scheduling problems. A problem common to truckload carriers, as well as rail and container operations, is dynamic fleet management. The basic components of the problem are a fleet of vehicles and other equipment along with information on current and forecasted customer demands. Decisions must be made with regard to the allocation of resources to meet current demands and anticipated future requirements. The vehicles and other related resources are initially distributed across a network of locations and decisions must be made now regarding the redistribution of the vehicles. Vehicles may be held in their current positions, repositioned empty in anticipation of future demand requirements, or moved loaded in order to service current demands. Typical applications can involve thousands of vehicles that need to be managed and thousands of loads that require service.

Information regarding customer demands and fleet status is continuously changing, and as such the motor carrier must be able to respond quickly and accurately. The inherent competitive nature of the motor carrier industry along with recent regulatory changes have made for an environment where efficient utilization of resources is critical [7]. Newly developed communications and information technology in conjunction with operations research techniques can be utilized to provide real-time decision support for operations managers.

1.1 Driver Assignment

Within the broad framework of dynamic fleet management are several subclasses of problems. One such subclass is the driver assignment problem. The primary decision is to assign drivers to loads at a minimum cost while adhering to all operational restrictions and customer requirements. Typically the drivers and loads are distributed throughout a network of locations and each driver is assigned to service an individual load. A common objective is to minimize the amount of empty miles travelled by the drivers in order to service the loads. Constraints may include time windows for the start of service, customer/driver preferences, and other operational constraints. An example driver assignment problem and solution is displayed in figure 1. The driver assignment problem is relatively simple when compared to other routing and scheduling problems. However, the simplicity of the problem does not undermine its importance.

1.2 Dynamic Assignment Problem

The primary focus of the paper is an extension of the driver assignment problem known as the driver scheduling or dynamic assignment problem. Drivers/vehicles are assigned to a sequence of tasks over a planning horizon that ranges from 24 hours to several days. The solution to the problem is a set of tours for the vehicles that determines the course of action for every individual vehicle over the planning horizon. The tours are designed to meet a number of objectives which may include cost minimization/profit maximization, customer satisfaction (e.g. on-time delivery), and driver satisfaction (e.g. balanced workload).

The complexity of the problem is amplified by the complicated work rules that govern the construction of tours and the costs associated with the tours. For example, there are regulations regarding the number of hours on the road and on duty a driver can accumulate in a given day. Collective union agreements may require guarantees with regard to hours worked and driver preferences for certain routes. There are also constraints regarding the service of the tasks. For example, if the task is comprised of moving freight between an O/D pair, there may be time window constraints for both the origin and destination, equipment

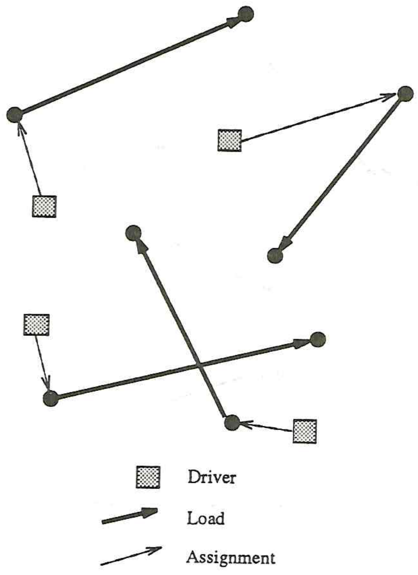


Figure 1: Driver Assignment Problem

requirements, and customer preferences with regard to the driver and vehicle type. The large number and intricacy of the driver work rules, complicated cost functions, and the size of most operations culminates to make the dynamic assignment problem very difficult. An example dynamic assignment problem is depicted in figure 2.

Once again the competitive nature of the industry and the demands of customers for quality service at the lowest possible cost require motor carriers to efficiently manage their operations. Operations research techniques combined with advances in computer and communications technology can be very valuable tools for managers faced with this difficult dynamic routing and scheduling problem.

The DAP is a general framework that can be used to model many transportation and logistics problems. Obvious motor carrier applications include long-haul full-truckload trucking and short-haul intermodal drayage operations. Many inventory distribution problems may also be described with the DAP. For example, bulk chemical manufacturers faced with the problem of maintaining customer inventories of various products must efficiently manage their fleet of delivery vehicles and trailers. Operations managers must design vehicle routes that distribute the necessary amounts of products to their customers while adhering to the many operational constraints regarding the construction of the tours. Dynamic crew scheduling problems provide another possible application for the DAP.

The successful implementation of a DAP solution strategy requires the ability to address real-time changes in the problem data. For example, new customer demands may appear during the course of the planning horizon, or a vehicle may be delayed due to equipment failure. The current vehicle tours must be re-evaluated when new information becomes available in order to optimize the given objectives. The solution strategy must produce good solutions efficiently to provide the fast response time necessary in a real-time environment. Therefore, this paper focuses on developing a solution strategy that produces high quality solutions, is flexible with regard to incorporating real-world problem details, and is amenable to a real-time implementation.

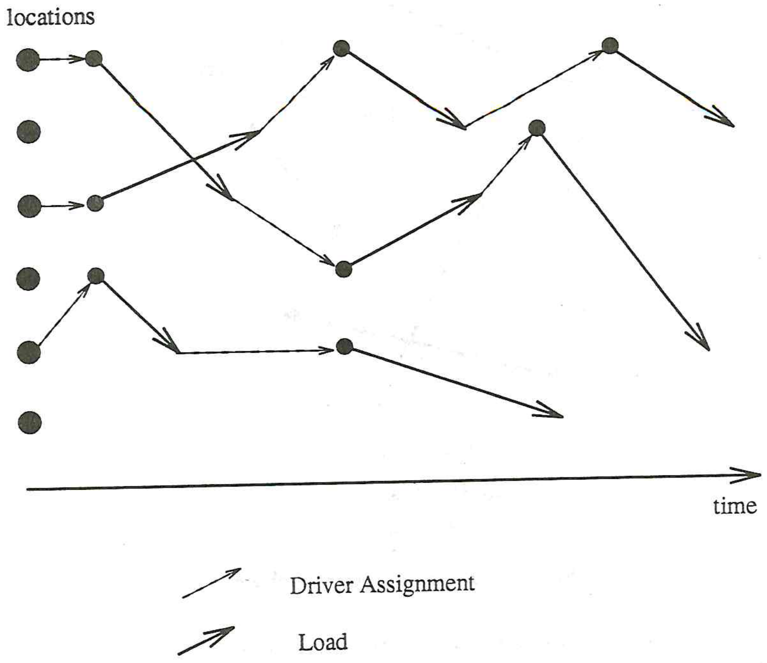


Figure 2: Dynamic Assignment Problem

1.3 Paper Contributions

The wide variety of applications for dynamic routing and scheduling problems and the important role they play in logistics and transportation were the motivating factors for their study in this paper. There clearly exists a need for improved understanding of the nature of the problems, and the development of efficient solution strategies. The latter is particularly important with regard to real-time implementations of dynamic routing and scheduling problems.

The paper concentrated on the solution of the dynamic assignment problem, a flexible modelling strategy that can be applied to many types of dynamic routing and scheduling problems. A label setting rolling horizon heuristic algorithm (LASER), combining components of optimization and simulation, was developed and implemented to solve the model. An extensive computational study was undertaken to provide a basis for evaluating the performance of the LASER algorithm. The LASER algorithm proved to be effective from both a computational and solution quality standpoint. The algorithm was capable of finding solutions within 1% of the LP-optimal solution for a number of randomly generated and real-world test problems. For problems that were tightly constrained with respect to the number of available vehicles, the algorithm's performance was poor; the LASER solutions were within 10% of the LP-optimal solution. The poor performance of the algorithm for the constrained problems was primarily due to its inability to efficiently utilize the available vehicle duty time.

The remainder of the paper is organized as follows: dynamic assignment problem description, LASER development, computational study, discussion of results, and conclusions.

2 Dynamic Assignment Model

There are several dynamic routing and scheduling problems that can be described by the dynamic assignment model. The following section describes the dynamic assignment problem from the context of a motor carrier application. The latter section then provides a mathematical formulation for the dynamic assignment model.

2.1 Problem Description

The dynamic assignment problem arises frequently in the context of fleet management for motor carriers. We will use a motor carrier application to motivate the development of the dynamic assignment problem. There exists a fleet of vehicles that are to be scheduled to service a set of known tasks. Each vehicle starts at a known location with a known time of availability. The set of tasks (loads to be transported between specified origins and destinations), and all pertinent task information, is assumed to be known at the beginning of the planning horizon. The goal is to construct minimum cost (maximum profit) tours for each vehicle subject to a number of constraints.

Every task is assumed to be a full trailer with every vehicle having the capacity of one trailer (no freight consolidation). Every task must be serviced by no more than one vehicle. Each task has a specified time window $[a_i, b_i]$ in which it must be picked up at the origin. The time windows are assumed to be hard; a vehicle is allowed to arrive before the start of the time window but must wait until a_i before starting service, and a vehicle is never allowed to arrive and start service after b_i. In the soft time window case vehicles would be allowed to arrive outside of the time window with a corresponding early/late penalty. The service time for each task is a combination of the load/unload times and the loaded travel time between the origin and destination. The length of every tour is limited by a known maximum duty time for each vehicle. Every tour must return the vehicle to its domicile without violating the duty time restrictions. A vehicle's initial location and domicile are not assumed to be the same physical location. It is not necessary to satisfy all customer demands; uncovered tasks may be viewed as refused loads or loads given to a common carrier.

We have made a number of simplifying assumptions that are not necessarily true for most real-world applications. For example, we have assumed a homogeneous fleet of vehicles and human operators. In addition, we have chosen to ignore equipment requirements for tasks. These and other simplifying assumptions were made to keep the model as clean as possible from a mathematical standpoint. In the following sections we describe how different real-world problem features can be addressed using the LASER algorithm.

For most real-world applications the assumption of a static problem is incorrect. Customers call in new tasks during the course of the planning horizon and the vehicle tours must be reconstructed. As mentioned in Chapter 2, one cannot simply insert new tasks into existing tours and guarantee that the solution will remain optimal. The definition of an optimal solution is not even clearly defined in a real-time setting. For the purposes of this paper we will assume all problem information is known at the beginning of the planning horizon. The LASER algorithm is amenable to reoptimizing as new data becomes available for a real-time implementation. The next section formulates the DAP described above as a mixed integer program.

2.2 Formulation

The dynamic assignment problem can be modelled as a network with complicating constraints. Consider the following notation for the DAP described above:

- D - set of vehicle nodes (source nodes)
- L - set of task nodes (transshipment nodes)
- $\{H\}$ - vehicle domicile node (sink node)
- $V = D \cup L \cup \{H\}$ - complete set of nodes
- A_k - set of directed arcs from vehicle node $k \in D$ to nodes $L \cup \{H\}$

- $A = \cup_{k \in D} A_k$ - complete set of vehicle arcs
- B_L - set of directed arcs between task nodes
- B_H - set of directed arcs between task nodes and the domicile node.
- $B = B_L \cup B_H$ - complete set of task arcs
- $E = A \cup B$ - the complete set of arcs for the network.
- c_{ij} - cost for arc $(i, j) \in E$.
- t_{ij} - travel time for arc $(i, j) \in E$
- s_i - service time for node $i \in V$
- $[a_i, b_i]$ - service time window for node $i \in V$

Note:

1. For arcs $(i, j) : j \in L$, the cost c_{ij} is equal to total operating costs (travel cost, wages, etc.) minus task revenue. For arcs (i, H), the cost is equal to operating costs.
2. Service time s_i is equal to the combination of the load time at the origin, unload time at the destination, and loaded travel time between the origin and destination for $i \in L$, and zero otherwise.
3. For $k \in D$ the service time window is equal to $[T^o, T^o]$, where T^o is the time that the vehicle is first available. For domicile node H the service time window is equal to $[T^o, T^o + \delta]$, where δ is the maximum duty time for a vehicle.

The variables are binary flow variables and continuous time variables:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ flows on arc } (i, j) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$T_i = \text{ time that a vehicle arrives at node } i \tag{2}$$

Given the above problem information and notation the dynamic assignment problem may be formulated as the following mixed integer program:

$$\min \sum_{k \in D} \sum_{(i,j) \in A \cup B} c_{ij} x_{ijk} \tag{3}$$

s.t.

$$\sum_{k \in D} \sum_{i \in D \cup L_j} x_{ijk} \leq 1 \qquad \forall j \in L \tag{4}$$

$$\sum_{(i,j) \in A_k} x_{ijk} = 1 \qquad \forall k \in D \tag{5}$$

$$\sum_{(i,j) \in A_k \cup B_H} x_{iHk} = 1 \tag{6}$$

$$\sum_{k \in D} \sum_{i \in D \cup L_j} x_{ijk} - \sum_{k \in D} \sum_{i \in \{H\} \cup L_j} x_{jik} = 0 \quad \forall j \in L \tag{7}$$

$$a_i \leq T_i \leq b_i \qquad \forall i \in V \tag{8}$$

$$T_i + s_i + t_{ij} - T_j \leq (1 - x_{ijk}) M \qquad \forall (i, j) \in E, \quad k \in D \tag{9}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall (i, j) \in E, \quad k \in D \tag{10}$$

Note:

- $L_j = L \setminus \{j\}$
- $M =$ large constant
- $T_i = T^o$ for $i \in D$

The objective is to minimize total costs over the entire network. Constraint set (4) ensures that every task is covered by at most one vehicle. Constraint sets (5) and (6) ensures that every vehicle tour starts at the vehicle's initial location and terminates at the domicile node. Flow conservation at the task nodes is captured by constraints (7). The hard time window constraints are enforced by constraint set (8). The logical relationship between the flow variables and the time variables is represented by (9). The final set of constraints are binary constraints on the link flow variables. Constraints (8) and (9) eliminate subtours provided $t_{ij} + s_i > 0$.

The potentially large number of integer flow variables limits the use of mixed integer solvers for the DAP. A problem with 100 vehicles and 400 tasks would result in as many as 200,000 binary variables. The operational nature of the DAP requires a solution procedure that can solve the problem in at most a few minutes. To facilitate the necessary computational efficiency a label setting rolling horizon algorithm was developed that takes advantage of the underlying dynamic network structure of the DAP.

3 LASER Development

The dynamic assignment model presented above generally cannot be solved directly with the level of efficiency required for real-world implementations. The proceeding section describes a dynamic decomposition for the DAP that reduces it to a pure network problem. The latter section presents the development of the LASER algorithm which uses the decomposed network to solve the original DAP.

3.1 Dynamic Decomposition

One heuristic approach to solve the DAP is the repeated use of the classical assignment model. For each vehicle a set of assignment arcs to customer demands can be generated. Each assignment arc represents the repositioning of the vehicle from its current location to the load's origin. Only those arcs which are feasible are considered. For example, given that a vehicle is available at time t at location i, all arcs must satisfy the condition $t + t_{ij} \leq b$, where j is the task's origin and b is the latest time at which the load can be picked up. After solving the assignment model each vehicle's time and location of availability and remaining duty hours can be updated. Those demands which were covered in the first assignment can be removed from the problem. A new assignment problem can then be generated and solved based on the updated information. This process can be repeated until there are no more tasks to be covered. In this manner a complete vehicle tour is built over the planning horizon.

The greedy algorithm described above is myopic and tends to perform poorly when the planning horizon is long. The dynamic decomposition approach attempts to perform multiple assignments by implicitly updating the vehicle information and generating a complete vehicle tour in one iteration. In order to accomplish the above goal the concept of a potential resource is introduced.

We define a potential resource in the context of the DAP as a vehicle at the end of a task. Upon completion of every task a vehicle will become available to possibly be assigned to another task. However, the state of the potential resource is not completely known. For the DAP, the state of the potential resource would include time and location of availability along with remaining duty time. The location of availability is the destination of the task associated with the potential resource. Each task has a time window so one does not know when the potential resource associated with the task will become available. The amount of remaining duty time for the potential resource is also unknown. Note that the state of a vehicle and a potential vehicle contain the same elements.

One approach is to set the state of each potential resource to its most optimistic value. For example, for a potential resource associated with task i with time window $[a_i, b_i]$ and service time s_i, the optimistic time of availability would be $a_i + s_i$. Similar optimistic values could be used for other elements of the state variable. One could then generate an assignment problem with vehicles and potential vehicles and complete vehicle tours could be produced in one iteration. By tracing the solution to the decomposed dynamic assignment problem one can determine the identities of the potential vehicles. For example, if vehicle 1 is assigned to task A and potential vehicle A is assigned to task B, then the tour for vehicle 1 consists of task A followed by task B.

Consider the following notation for the decomposed DAP:

- V = set of vehicle nodes, $|V| = n$.
- T = set of task nodes, $|T| = m$.
- P = set of potential vehicle nodes, $|P| = m$.
- $\{D\}$ = dummy task (sink) node.
- $T' = T \cup \{D\}$, total set of task nodes.
- A = set of vehicle to task assignment arcs (includes feasible arcs only).
- B = set of potential vehicle to task assignment arcs (includes feasible arcs only).
- C = set of task to potential vehicle arcs (demand arcs).
- c_{ij} = cost of assigning vehicle i to task j.
- d_{ij} = cost of assigning potential vehicle i to task j.
- r_j = net contribution associated with task j.

The cost of assigning a vehicle or potential vehicle to a task is the total operating costs associated with relocating the vehicle from its current position to the origin of the task. The net contribution for a task is equal to revenue minus operating costs associated with covering the task (usually a negative quantity). The arcs from the real/potential vehicles to tasks represent empty movements while task to potential vehicle arcs represent loaded movements. Arcs into the dummy task node represent vehicles returning to their domicile. Feasible potential vehicle to task arcs are generated based on the most optimistic state for the potential vehicle.

There are three types of integer flow variables for the decomposed DAP:

$$x_{ij} = \begin{cases} 1 \text{ if vehicle } i \in V \text{ is assigned to task } j \in T' \\ 0 \text{ otherwise} \end{cases} \tag{11}$$

$$y_{ij} = \begin{cases} 1 \text{ if potential vehicle } i \in P \text{ is assigned to task } j \in T' \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

$$z_j = \begin{cases} 1 \text{ if task } j \in T \text{ is covered} \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

The decomposed DAP can then be formulated as the following pure network problem:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} + \sum_{(i,j)\in B} d_{ij} y_{ij} + \sum_{j\in T} r_j z_j \tag{14}$$

s.t.

$$\sum_{j\in T'} x_{ij} = 1 \qquad \forall i \in V \tag{15}$$

$$\sum_{j \in T'} y_{ij} = 1 \qquad \forall i \in P \qquad (16)$$

$$\sum_{i \in V} x_{ij} + \sum_{i \in P} y_{ij} - z_j = 0 \qquad \forall j \in T \qquad (17)$$

$$\sum_{i \in V} x_{ij} + \sum_{i \in P} y_{ij} = n \qquad j = D \qquad (18)$$

$$x_{ij} \geq 0, \text{ integer} \qquad \forall (i,j) \in A \qquad (19)$$

$$y_{ij} \geq 0, \text{ integer} \qquad \forall (i,j) \in B \qquad (20)$$

$$z_j \text{ binary} \qquad \forall j \in T \qquad (21)$$

The objective function (14) represents cost minimization. Constraint sets (15) and (16) ensure that real or potential vehicles are assigned to exactly one task (including the dummy task). Constraints (17) restrict each task to be covered by at most one real/potential vehicle. The last three sets of constraints are the bounds and integrality constraints for the flow variables. The decomposed DAP is a pure network problem that can be solved efficiently by using a network simplex code. See Figure 3 for an example decomposed DAP network. Figure 4 illustrates a solution to the decomposed DAP and the corresponding vehicle tours.

By assuming the potential resources have the most optimistic states, the time window and length of tour constraints have been implicitly relaxed. Therefore, the solution to the decomposed DAP may result in tours which are infeasible. For example, if the time window for task A is between 11:00 and 4:00 with a service time of 60 minutes, the potential vehicle is expected to be available at 12:00. When the entire vehicle tour is traced it may result in the real vehicle assigned to task A not completing the task until 3:00. The fact that the vehicle was available at 3:00, not 12:00, may make the next leg in the tour infeasible, i.e. the vehicle would have to leave sometime before 3:00 to satisfy the time window constraints for the next task in the tour. The same type of scenario can happen with respect to length of tour constraints.

For problems with perfectly tight time windows and no tour length limits, the decomposed dynamic assignment model and the original DAP are equivalent. The presence of time windows and finite tour length limits results in the decomposed DAP being a relaxation of the original DAP. Another drawback of the decomposed DAP is the possible existence of subtours in the solution. Depending on the structure of the problem it is possible for a solution to contain cycles (e.g. potential vehicle A assigned to task B and potential vehicle B assigned to task A). However, if the solution to the decomposed DAP does not violate any of the time windows, tour length constraints, or contain subtours, then the solution is optimal for the DAP.

In most applications the use of the optimistic state variables for the potential vehicles will result in a solution with infeasible tours. The question of how to use the concept of potential resources to construct good feasible tours then arises. The following section describes the LASER algorithm which uses the potential resource concept to construct vehicle tours.

3.2 LASER Algorithm

The decomposed DAP is a pure network and can be solved efficiently using commercial network simplex code. The primary disadvantage is the uncertainty in estimating the potential resource's state can lead to highly infeasible solutions. The LASER algorithm is a tour construction procedure that builds tours based on a rolling horizon using the concept of potential resources. The following section describes the label setting rolling horizon algorithm.

For every potential resource in the network there is an associated label which contains the following elements:

1. expected time of availability (ETA)

2. ETA location

3. available duty time (ADT)

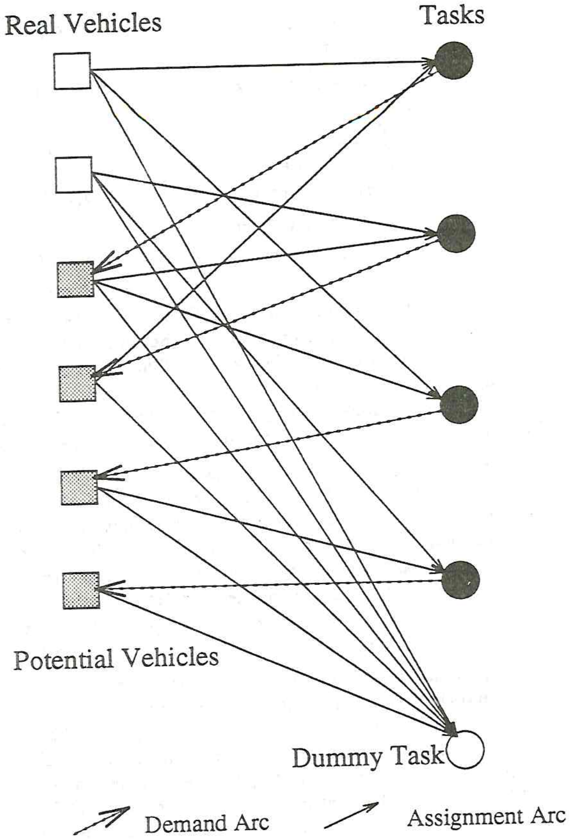


Figure 3: Decomposed DAP Network

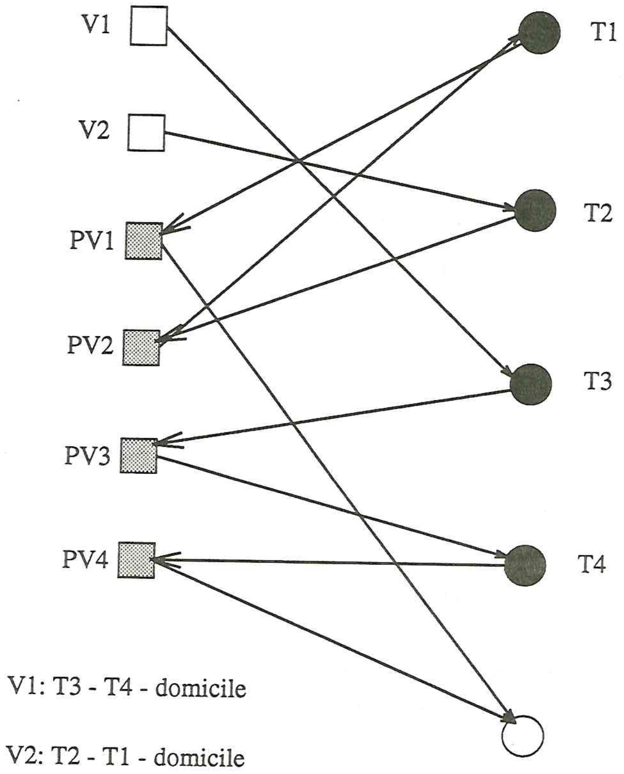


Figure 4: Solution to DAP Network

4. status

5. other attributes

The first three elements constitute the state variable for the potential vehicle. The status of the label may be any one of three values: undefined, temporary, or permanent. A potential vehicle with an undefined status cannot be assigned to any tasks because the elements of the state variable are too uncertain. A potential vehicle with temporary status can be assigned to tasks, but the state variable elements are subject to change. Finally, a potential vehicle label has permanent status if all the elements of the state variable have been fixed. A potential vehicle with permanent status is equivalent to a real vehicle. For a potential vehicle's state variable to be fixed the path (tour) that includes that task must be fixed. We now formally present the LASER algorithm.

- *Step 0:* INITIALIZE
  - Get T^o and T^f, the start and end of the planning horizon respectively and set $T = T^o$.
  - Get $\delta > 0$ and $\epsilon > 0$ with $\delta \geq \epsilon$.
  - Initialize all potential vehicle labels by setting the state variable elements to the optimistic values and the status to UNDEFINED.
  - Build the initial decomposed DAP network.
  - NUMTEMP = 0
- *Step 1:* SOLVE THE NETWORK
  - Solve the decomposed DAP network.
  - Note: A potential vehicle cannot be assigned to a task if its label status = UNDF.
- *Step 2:* UPDATE POTENTIAL VEHICLE LABELS
  - Check for cycles: if a task is covered on a subtour set the label status of its corresponding potential vehicle to UNDF.
  - Set $T' = T + \delta$
  - Simulate each acyclic vehicle tour, calculate NEW-ETA and NEW-ADT for each potential vehicle.
  - For each potential vehicle with label status $\neq$ PERM do:
    - If NEW-ETA $\leq T'$ and the tour preceding the PV is feasible, then set status = TEMP, ETA = NEW-ETA, and ADT = NEW-ADT.
    - Else if status = TEMP and NEW-ETA $> T'$, then set status = UNDF.
  - Update NUMTEMP.
- *Step 3:* UPDATE PERMANENT LABELS
  - *Step 3.1:* Set T_{min} = minimum ETA over all potential vehicles with label status = TEMP ($T_{min} = \infty$ if NUMTEMP = 0).
  - *Step 3.2:* If $T_{min} \leq T + \epsilon$, then $\bar{T} = T_{min}$, for all potential vehicles with TEMP label status and ETA $\leq \bar{T}$ set label status = PERM, update NUMTEMP and go to Step 3.1.
  - If NO labels were set to PERM above and $T < T^f$, then set $T \leftarrow T + \epsilon$. Go to Step 4. Else stop.
  - If $\bar{T} < T^f$ then set $T = \bar{T}$. Go to Step 4. Else stop.
- *Step 4:* ADJUST NETWORK
  - For potential vehicles with status = PERM fix flow on the associated path.
  - For potential vehicles with status = TEMP recalculate the feasibility and cost for arcs to task nodes given the potential vehicle's current state variable.

- For potential vehicles with status = UNDF, restrict flow to the domicile node.
- Go to Step 1.

The initial network is equivalent to the network in the decomposed DAP with the potential vehicle state variables set to their optimistic values. However, initially the potential vehicle labels have an undefined status so that flow is restricted to the domicile node only (i.e. $y_{ij} = 0, \forall j \in T$). Therefore, the first network that is solved is identical to solving a simple vehicle to task assignment problem.

Step 2 examines the current solution and determines which potential vehicles will have their status changed from undefined to temporary or vice- versa. Every vehicle tour is simulated to determine the state of the potential vehicle associated with every task on the tour. Suppose $T' = 11{:}00$ and vehicle 1 has the following tour $\{A \rightarrow D \rightarrow C \rightarrow \text{domicile}\}$, with the following label status values PV_A = TEMP, PV_D = TEMP, and PV_C = UNDF. Simulation of the vehicle tour results in the following data:

- PV_A: NEW-ETA = 10:00 NEW-ADT = 10 hrs. tour = FEASIBLE
- PV_D: NEW-ETA = 12:00 NEW-ADT = 8 hrs. tour = FEASIBLE
- PV_C: NEW-ETA = 16:00 NEW-ADT = 4 hrs. tour = INFEASIBLE

Given the results of the tour simulation and the current value of T', PV_A would have its state variable updated to the new values and its status would remain temporary. The label for PV_C would have its status changed from temporary to undefined because its NEW-ETA is greater than T'. The label for PV_D would remain unchanged because its NEW-ETA is greater than T' and the tour is infeasible at this point.

The next step determines which potential vehicles will have their label status changed to permanent. Continuing with the example above, suppose $\bar{T} = 10{:}00$. The status for PV_A would be changed to permanent resulting in the assignment of vehicle 1 to task A being fixed. If the end of the planning horizon has not been reached then the network is adjusted. The algorithm may be described as a window that is rolled forward at each iteration, with δ being the width of the window and ϵ the minimum advancement (see figure 5). Potential vehicle labels with ETAs beyond the window have an undefined label status. Labels of potential vehicles within the window may have either temporary or undefined status depending on the feasibility of the tour associated with the potential vehicle. As the window rolls forward potential vehicles with temporary label status are set to permanent.

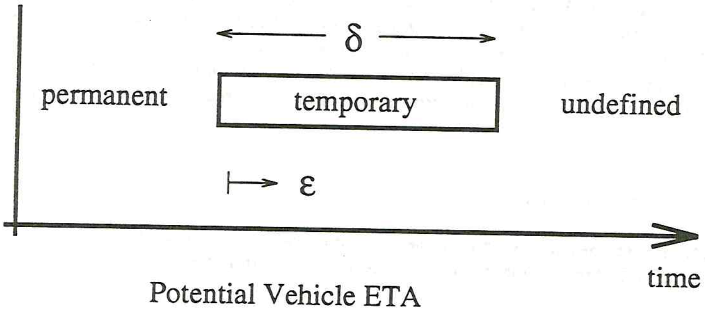


Figure 5: Rolling Window

The adjustment of the network is conducted based on the new values for the potential vehicle labels. If the status has been set to permanent then the path that preceded the potential vehicle is fixed. In the above example this would result in x_{1A} being set to one. Note that when a potential vehicle node has its label status set to permanent, the label status for all preceding potential vehicles nodes on that path have permanent label status. For potential vehicle nodes with temporary label status the arcs to task nodes are re-evaluated. If the arc (i, j) is infeasible based on the new state variable for PV_i, then y_{ij} is set to zero. Otherwise the value of d_{ij} is updated based on the new state at PV_i. Those potential vehicle nodes with undefined label status have $y_{ij} = 0, \forall j \in T$. The network is then resolved based on the recent adjustments and the above procedure is repeated.

The end of the planning horizon is reached in a finite number of steps; the maximum number of steps required is $(T^f - T^o)/\epsilon$. At the end of the planning horizon all of the potential vehicle labels will be either permanent or undefined. The tasks associated with potential vehicles with undefined label status are uncovered tasks. Depending on the specific application, these tasks may be viewed as refused loads or loads that will be given to a common carrier. Tasks may be refused because there were not enough vehicles or because the revenue associated with the load was not enough to offset the operating expenses of covering the task. The other tasks are covered by feasible vehicle tours.

The LASER algorithm described above is very flexible with respect to incorporating additional constraints. Suppose there are two types of vehicles, V_1 and V_2, and each task specifies whether it can be serviced by V_1, V_2, or both. We would then add a vehicle type to the state variable for the potential vehicles. The vehicle type would be uncertain for potential vehicles corresponding to tasks that could be serviced be either V_1 or V_2. We could use the same label updating mechanism described above to address equipment requirement constraints.

The LASER algorithm is also well suited for real-time implementations which are particularly important for most real-world applications. Information updates can be incorporated by modifying the existing network and re-executing the LASER algorithm. For example, if a new task is called in by a customer it would simply be appended to the current network by adding the necessary task and potential vehicle nodes and the appropriate arcs. If an equipment failure occurs and results in a vehicle being available at a later time than expected, the driver to task arcs would then be re-evaluated in terms of feasibility and cost. Information from the previous solution can be used when resolving the problem. For example, parts of a entire tour can be fixed by keeping the appropriate potential vehicle status labels on the path set to permanent.

4 Computational Study

The following sectio contains a computational study and evaluation of the LASER algorithm. The algorithm is compared to a shortest path tour construction procedure (SPTC) and a Dantzig-Wolfe decomposition column generation procedure (DWCG). The DWCG solution method is used to provide a lower bound on the optimal solution and the SPTC algorithm provides an upper bound. The size of the rolling window and the rate at which it advances are also studied to see how they impact on the algorithm's performance. The algorithms are tested on randomly generated problems and problems from a real-world motor carrier. The algorithm analysis uses a number of evaluation criteria including cpu time, and different measures of solution quality.

4.1 Algorithm Implementation

The following section describes issues that arose in implementing the LASER algorithm. The dynamic programming based SPTC algorithm is described in section 4.1.2 and the DWCG procedure is described in section 4.1.3.

LASER There are a number of implementation issues that arise in the use of LASER. The first major issue is the generation of the network. For large problems explicitly enumerating all the possible real/potential vehicle to task arcs would be too time consuming. We implemented a spacefilling curve procedure for finding a candidate set of arcs for the real/potential vehicle nodes (see Bartholdi and Platzman [1]). The procedure added arcs to the network by first finding the n closest tasks for each vehicle and then finding the n closest vehicles for each task. The closeness of a task to a vehicle is determined by their relative positions on the spacefilling curve.

The optimal solution to the network was determined by using Simpnet, a network simplex code (see [6] and [5]). Simpnet is capable of solving relatively large pure network problems in a few seconds on a typical computer workstation. All of the algorithms were coded in ANSI C and compiled and executed on a Silicon Graphics Indy workstation.

Shortest Path Tour Construction In order to evaluate the performance of LASER we implemented a shortest path based tour construction procedure. The SPTC procedure determined vehicle tours by finding the least cost path from a vehicle node to its domicile node using the same network generated for LASER. The algorithm is based on the shortest path label correcting algorithm of Glover et al. [4] and the shortest path with time windows algorithm of Desrochers and Soumis [3], [2]. The algorithm determines the least cost path from the vehicle node to the domicile node while satisfying the time window and length of tour constraints (see [8]).

The SPTC algorithm is as follows:

- *Step 0:* INITIALIZE
  - Generate the decomposed DAP network.
  - $k = 1$
- *Step 1:* CONSTRUCT VEHICLE TOUR
  - Determine the least cost path for vehicle k.
- *Step 2:* UPDATE
  - Remove the tasks covered on the path from the network.
  - If there are still tasks to be covered and k is less than the number of vehicles then $k \Leftarrow k + 1$, Go to Step 1. Otherwise stop.

The SPTC algorithm outlined above is dependent on the order in which the vehicles are treated. For example, if there are ten vehicles with ADT = 20 and ten with ADT = 40, the solution will vary based on the order in which the vehicle tours are constructed. Nonetheless, the SPTC algorithm provides a good solution to the DAP.

Dantzig-Wolfe Decomposition One possible solution approach for the DAP is to enumerate different feasible paths through the network to develop a set of candidate routes for each vehicle. One could then choose the best route for each vehicle. After choosing the best routes one might ask if there are better routes that were not considered in the candidate set. This is precisely the framework in which Dantzig-Wolfe decomposition with column generation operates. The DWCG procedure implemented for the DAP is based on the vehicle routing problem with time windows work of Desrosiers et al. [3].

The DAP can be easily transformed into a problem which is well suited for a Dantzig-Wolfe decomposition column generation scheme. The restricted master problem considers a set of feasible routes for each vehicle and selects the optimal subset of paths such that each vehicle is assigned to exactly one path.

The restricted master can be formulated as:

$$\min \sum_{k\in D} \sum_{p\in\Omega_k} c_{pk}\theta_{pk} + \sum_{i\in T} r_i s_i \tag{22}$$

s.t.

$$\sum_{k\in D} \sum_{p\in\Omega_k} a_{ipk}\theta_{pk} + s_i = 1 \qquad \forall i \in T \tag{23}$$

$$\sum_{p\in\Omega_k} \theta_{pk} = 1 \qquad \forall k \in D \tag{24}$$

$$s_i \text{ binary} \qquad \forall i \in T \tag{25}$$

$$\theta_{pk} \text{ binary} \qquad \forall p \in \Omega_k, \ \forall k \in D \tag{26}$$

where:

- Ω_k - set of candidate paths (tours) for vehicle k
- c_{pk} - cost of path p for vehicle k
- r_i - net contribution for task i
- a_{ipk} - 1 if task i is on path p for vehicle k, 0 otherwise
- s_i - slack variable
- θ_{pk} - 1 if path p is utilized for vehicle k, 0 otherwise

The above formulation is a set partitioning problem with additional generalized upper bound (GUB) constraints for each vehicle. The binary constraints can be relaxed and replaced with linear constraints $0 \leq \theta_{pk} \leq 1$ to form a linear relaxation. Let α_i be the dual variables associated with the set partitioning constraints, and let β_k be the dual variable associated with the resource constraint.

The subproblem for this formulation is the same shortest path problem solved in the SPTC algorithm. Note that a shortest path problem must be solved for each vehicle and that these problems are independent. The dual variables from the restricted master problem are used to calculate reduced costs on arcs in the network. The marginal cost $\bar{c}_{pk}$ of path p for vehicle k is given by:

$$\bar{c}_{pk} = c_{pk} - \sum_{i \in T} \alpha_i a_{ipk} - \beta_k \tag{27}$$

The DWCG solution procedure used the same network as the LASER and SPTC algorithms. The dual information from the master can be used to calculate the reduced costs for arcs in the decomposed DAP network.

The subproblem can be solved efficiently by using the same shortest path algorithm used for the SPTC procedure. The shortest path algorithm can generate several negative reduced cost columns at each iteration. The DWCG procedure can be accelerated by adding several columns at once for each vehicle subproblem. In the actual implementation an upper limit was set on the number of columns to add for each subproblem. If no negative reduced cost columns can be found then the solution procedure terminates and the current solution provides a lower bound on the optimal integer solution. Experience has shown that this lower bound is tight.

4.2 Problem Sets

The test problems consisted of two classes of DAPs: randomly generated problems and real-world problems. The main difference between the problems is the spatial distribution of the demand movements. The real-world problems contain a central terminal which is either the origin or destination of all the demands, whereas the O/D pairs are uniformly distributed in the generated problems. The following section describes the methodology used to create the generated problems, and the latter section describes the real-world data sets.

Generated Test Problems In order to evaluate the performance of the LASER algorithm a number of randomly generated problems were created. To create the test problems we used a problem generator to simulate customer demands for the DAP. The problem generator used a non-homogeneous Poisson process described by a weekly demand rate parameter with day-of-week and hour-of-day distributions. In this manner the number of loads generated per week is dependent of the current time and day of week. There are several components needed to describe a load: call-in time, origin/destination, pickup time, and the width of the pickup window.

A non-homogeneous Poisson process was used to generate the call-in times. For our testing purposes we were interested in solving a DAP with a planning horizon of one day, with all the demands known at the start of the day. Therefore, we specified the day-of-week and time-of-day distributions so that all demands would be called-in at the start of our planning horizon. The origin/destination pair for a demand was generated uniformly from a given set of locations. The pickup time for the demand was generated from a specified discrete distribution for the hour of day. The minutes after the hour was then chosen from a uniform distribution. The width of the pickup time window was generated from a discrete distribution and appended to the pickup time.

Using the problem generator outlined above we created nine test problems. The test problems are characterized by the number of tasks and the distribution of the time windows. We generated problems with 50, 75, and 100 tasks with tight, wide, and mixed time windows. The width of a wide time window is equal to the length of the planning horizon. In the mixed time window case we used a distribution where 25% of the windows were tight, 25% were wide, and the remainder were uniform between tight and wide. The set of locations was taken from the customer data base for the drayage problem described in the next section.

Drayage Test Problems A specific application of the DAP is the drayage problem. The problem arises in intermodal operations where trains and trucks are used to transport freight. Typically a drayage company truck will pickup the freight (container) from the shipper and transport it to the nearest rail terminal. The container is then loaded onto a train and transported to the rail terminal closest to the load's final destination. Finally the freight is moved by truck to the consignee. An intermodal operation will consist of a few geographically dispersered rail terminals each with a fleet of trucks for the pickup and delivery of freight.

The problem is to develop a set of minimum cost driver tours over a planning horizon which is typically 24 hours. Every load has a time window specifying the earliest and latest time at which the load may be picked up or delivered. The time window constraints are hard (i.e. early arrivals are allowed but service cannot begin until the start of the window, and late arrivals are not allowed). There are other constraints including driver work rules and equipment requirements. In addition to using its own fleet of vehicles, the drayage company may use trucks from other companies (at a premium cost) to cover loads. Due to the nature of the drayage problem dispatchers manually construct tours by finding triangulation opportunities. A typical tour is: deliver freight from the rail terminal, move empty to pickup freight, deliver freight to the rail terminal. The rail terminal behaves as a regeneration point.

The fact that most tasks originate or terminate at the rail terminal differentiates the drayage problem from the randomly generated problems. Other features like multiple loads between the same O/D pair with the same characteristics also make the drayage problem different. The naturally occurring degeneracy of the drayage problem makes it a difficult problem class to solve with the DWCG procedure. We used three different real-world drayage sets. The drayage problems contain tasks with a mixed time window distribution similar to the one used in the problem generator.

4.3 Discussion of Results

The subsequent sections discuss the results of the numerical study conducted to evaluate the LASER algorithm. The first section describes the effects of the time window parameters δ and ϵ on solution quality and cpu time. The second section contains the comparison and analysis of the three algorithms used to solve the DAP.

LASER Parameters The LASER algorithm has two user defined input parameters that affect the potential vehicle label setting procedure. The width of the rolling window is specified by δ, and ϵ specifies the minimum advancement of the window at each iteration of the algorithm. The width of the time window relative to the length of the planning horizon affects the number of potential vehicle labels considered for temporary status per iteration. For small values of δ the labels of potential vehicles in the future will remain undefined, whereas for large values of δ more labels will be set to temporary earlier in the algorithm. As

100 tasks, 30 vehicles		$\epsilon = 60$	$\epsilon = 30$	$\epsilon = 15$	$\epsilon = 1$
$\delta = 60$	Profit	77,192	78,181	80,742	77,500
	cpu time	3	3	5	12
$\delta = 120$	Profit	76,416	79,686	81,382	83,324
	cpu time	3	4	6	11
$\delta = 180$	Profit	78,961	82,414	81,682	82,453
	cpu time	3	5	6	11
$\delta = 360$	Profit	80,707	84,912	84,724	84,182
	cpu time	4	7	9	18
$\delta = \infty$	Profit	83,493	86,777	87,411	87,455
	cpu time	5	10	13	25

Table 1: Rolling Window Parameters

mentioned earlier, the number of iterations required for the algorithm is bounded by $(T^f - T^o)/\epsilon$, therefore the amount of cpu time is inversely proportional to ϵ. The smaller the value of ϵ the fewer potential vehicles labels set to permanent status per iteration.

The combination of window parameters that will produce the best results is not obvious. The LASER algorithm was tested on a randomly generated problem with 100 tasks (wide time windows) and 30 vehicles with different values for δ and ϵ. The results of the experiment are contained in table 1. All recorded cpu times include network generation and algorithm execution but do not include I/O time.

As expected the amount of cpu time increased as ϵ decreased; the cpu time increased by four to five times as the value of ϵ was decreased from 60 minutes to one minute. The solution quality improved as the step size for the rolling window decreased. The cpu time and objective function (profit) increased as the width of the time window increased. The cpu times doubled as the δ was increased from 60 minutes to ∞ (the length of the planning horizon).

Based on the results in table 1 and other experiments it is clear that the best policy for the LASER algorithm is small ϵ and large δ. There is however a clear trade off between solution quality and cpu time. The results of these rolling window parameter experiments seem to indicate that even some information about a potential vehicle is better than none, and the tours should be built as slowly as possible to allow for changes in subsequent iterations. Given the above results, all subsequent problems solved with LASER had $\epsilon = 1$ and $\delta = \infty$.

Comparison of Algorithms The performance of the LASER algorithm was evaluated by comparing its solution characteristics to those of the SPTC and DWCG solution strategies for the test bed of generated and real-world problem described earlier. The following statistics were recorded for every problem:

- objective function value in dollars (profit)
- number of idle drivers (idle)
- number of tasks left uncovered (refusal)
- percentage of total miles travelled empty by the vehicles (% empty)
- percentage of total available vehicle duty time utilized (% util.)
- cpu time in seconds

The profit for a set of tours was calculated as the total revenue from covered loads minus total travel costs (loaded and empty miles). All of the algorithms are profit maximization based and do not directly address the other measures outlined above. The other measures are helpful for evaluating differences in the objective function between solutions. For example, two solutions may cover the same number of tasks but one has fewer empty miles and consequently a higher profit.

Every test problem was solved with two values for the number of vehicles. The two different fleet sizes were used to measure the performance of LASER under excess and constrained scenarios. A problem with excess vehicles is one in which all coverable tasks are assigned and there are idle vehicles. Conversely, a constrained problem is one in which there are no idle vehicles. The two fleet size scenarios combined with the generated and real-world test problems resulted in a total of 24 problems. The results for the generated problems are contained in tables 2, 3, 4, and the real-world drayage problem results are presented in tables 5, 6, and 7. Note that DWCG profit values marked with an * are for the optimal integer solutions and DNR denotes problems for which DWCG did not run. All other DWCG solutions were fractional and represent an upper bound on the maximum profit.

The vehicles used for testing purposes were all assumed to be homogeneous with respect to their characteristics (starting time and location, domicile, and available duty time). A homogeneous vehicle fleet was used for two reasons. Firstly, a homogeneous vehicle fleet eliminates the order dependence problems associated with the SPTC procedure. Secondly, in the DWCG procedure it is no longer necessary to solve a shortest path problem for every vehicle at each iteration of the algorithm. The GUB constraints were replaced with one constraint of the form $\sum_{p \in \Omega} \theta_p = n$, where n is the number of vehicles. Consequently the cpu times for the DWCG procedure are much less than they would be in the case of a heterogeneous vehicle fleet. The difference in cpu times may be as high as a factor equal to the number of vehicles.

The results of the experiments for both the generated and real-world problems provides explicit evidence on the ability of LASER to solve the DAP. The LASER algorithm performed well with respect to profit when the problem was loosely constrained with respect to fleet size. However, when there were fewer vehicles available to cover the same set of tasks the algorithm's performance deteriorated. The percentage gaps between the DWCG upper bound and the objective function values of LASER and SPTC for the generated problems are presented in table 8. The average gap for the excess vehicle problems is 0.8% indicating that LASER produced optimal or near-optimal solutions. The average percentage gap increased by an order of magnitude to 4.4% for the constrained versions of the same problems.

The SPTC solution procedure displayed the same behavior with slightly poorer solution quality than LASER. The average percentage gaps were 1.4% and 5.8% for the excess and constrained problems respectively. The SPTC procedure was superior to LASER with respect to cpu time for 23 of the 24 test problems. The SPTC procedure also used fewer vehicles than LASER to cover the same number of loads when there was excess fleet capacity.

To further investigate the performance of the algorithms for fleet size constrained DAPs, a 100 task, mixed time window, randomly generated problem was solved with ten different fleet size values. The objective function values are plotted in figure 6 and the percentage gaps are plotted in figure 7. The deterioration in LASER's performance as the problem became more constrained is clear from the results in the two figures. The objective function percentage gap ranged from less than 1% for the excess problems to 25% for the constrained problems.

The SPTC procedure was able to better utilize the available vehicle duty time when the problem was constrained which resulted in more tasks being covered and thus produced a more profitable solution (see figures 6 and 7). The plots in figure 8 show the vehicle utilization for the algorithms. The figure suggests that LASER does not produce efficient vehicle tours for either the excess or constrained fleet size scenarios. When there were excess vehicles LASER used more of the available vehicle time than the other procedures indicating unnecessary vehicle utilization. In the constrained cases the LASER algorithm had a high vehicle utilization measure but still covered fewer tasks.

To achieve the high degree of efficient vehicle utilization required a lot more computational effort for the DWCG procedure. The degree to which the problem is constrained with respect to fleet size appeared to have an important influence on the DAP's level of difficulty. The sometimes dramatic increase in cpu

	Excess $n = 20$			Constrained $n = 10$		
WIDE	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	45,708	45,298	45,804*	38,872	38,094	42,294
Idle	3	3	2	0	0	0
Refusal	0	0	0	8	9	4.1
% Empty	18.6	25.7	16.8	22.5	22.1	22.6
% Util.	64.4	69.2	60.4	94.3	88.1	98.0
cpu time	10	3	13	10	2	709

	Excess $n = 40$			Constrained $n = 30$		
TIGHT	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	39,462	39,468	39,548*	38,212	36,840	38,948*
Idle	3	6	4	0	0	0
Refusal	4	4	4	6	8	5
% Empty	38.0	37.9	37.1	37.7	36.2	36.8
% Util.	65.4	62.2	61.1	73.1	71.6	71.0
cpu time	9	2	2	9	2	2

	Excess $n = 30$			Constrained $n = 20$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	44,152	44,156	44,489*	43,500	41,964	44,479*
Idle	5	7	6	0	0	0
Refusal	1	1	1	2	4	1
% Empty	28.1	28.0	22.7	27.4	25.9	22.9
% Util.	69.4	64.7	62.3	90.0	84.5	87.5
cpu time	11	3	5	10	1	13

Table 2: 50 Task Generated Problems

times for the DWCG procedure when the problem was constrained provides further evidence for the above assertion. For the problem examined above, the cpu times for DWCG ranged from 50 seconds to 1,050 seconds as the fleet size varied from 50 to 5, whereas LASER's cpu times ranged from 50 to 20 seconds.

The results of the computational analysis provides a good insight to the nature of DAPs and the ability of the LASER algorithm to solve them. LASER is clearly limited when the problem is constrained with respect to fleet size. However, to achieve the superior results requires a substantial increase in computational effort. The final section describes the main conclusions from the paper and outlines future research opportunities.

5 Conclusions

The paper focused on developing a new solution strategy for an important class of dynamic routing and scheduling problems known as the dynamic assignment problem. The DAP has many applications ranging from motor carrier fleet management to airline crew scheduling. Surprisingly, there has been relatively little research done in this area despite the number of existing applications. Our main objective was to design a solution procedure that was able to incorporate real-world problem features, produce high quality solutions, and be amenable to a real-time implementation. To achieve these objectives we developed a generalized label setting rolling horizon algorithm.

	Excess $n = 30$			Constrained $n = 20$		
WIDE	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	67,834	67,218	68,087	66,292	63,780	68,092
Idle	5	5	5	0	0	0
Refusal	0	0	0	2	5	0
% Empty	19.2	25.6	16.2	20.0	21.5	16.1
% Util.	66.4	69.8	61.7	93.1	86.2	92.4
cpu time	30	6	49	31	5	317
	Excess $n = 50$			Constrained $n = 40$		
TIGHT	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	60,524	60,496	60,632*	60,488	58,353	60,601*
Idle	5	7	6	0	0	0
Refusal	7	7	7	7	10	7
% Empty	36.8	37.0	35.9	37.1	36.6	36.1
% Util.	65.6	64.0	61.9	76.4	73.5	73.3
cpu time	15	4	2	14	4	3
	Excess $n = 50$			Constrained $n = 30$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	66,275	66,345	66,964*	61,668	64,882	66,952
Idle	12	18	18	0	0	0
Refusal	1	1	1	7	3	1
% Empty	31.2	30.6	24.5	31.3	30.5	24.6
% Util.	65.4	57.3	54.5	91.0	88.9	87.3
cpu time	31	6	16	33	6	35

Table 3: 75 Task Generated Problems

	Excess $n = 40$			Constrained $n = 25$		
WIDE	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	90,234	89,627	90,655	81,865	81,076	90,496
Idle	11	7	9	0	0	0
Refusal	0	0	0	10	11	0
% Empty	15.8	20.7	12.0	18.4	17.5	13.4
% Util.	66.7	69.3	62.0	92.7	86.2	97.1
cpu time	74	12	118	79	10	4,411
	Excess $n = 65$			Constrained $n = 50$		
TIGHT	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	83,328	82,947	83,349*	77,552	74,188	77,556*
Idle	2	3	1	0	0	0
Refusal	6	6	6	14	18	14
% Empty	35.6	37.8	35.5	33.3	34.3	33.9
% Util.	70.0	69.3	68.0	74.1	72.7	70.9
cpu time	24	8	4	21	6	3
	Excess $n = 50$			Constrained $n = 30$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	86,896	87,324	87,710	76,751	80,357	86,448
Idle	6	11	6	0	0	0
Refusal	4	4	4	17	13	5
% Empty	32.6	29.3	26	28.1	24.2	30.2
% Util.	76.2	68.8	67.9	91.2	92.6	95.1
cpu time	48	13	48	52	11	455

Table 4: 100 Task Generated Problems

	Excess $n = 50$			Constrained $n = 30$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	3,789	3,695	3,804*	3,771	3,695	3,797
Idle	8	33	16	0	13	0
Refusal	10	10	10	10	10	10
% Empty	12.9	16.7	12.3	13.7	16.7	12.6
% Util.	28.8	23.1	23.7	44.4	37.8	39.7
cpu time	4	4	57	4	4	57

Table 5: 61 Task Drayage Problem

	Excess $n = 40$			Limited $n = 30$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	3,814	3,819	3,924	3,767	3,705	3,919
Idle	1	10	6	0	2	0
Refusal	12	12	12	12	14	12
% Empty	32.5	32.4	30.4	33.4	32.7	30.5
% Util.	44.8	42.8	39.5	57.0	55.2	51.6
cpu time	5	4	292	5	4	292

Table 6: 73 Task Drayage Problem

	Excess $n = 50$			Limited $n = 30$		
MIXED	LASER	SPTC	DWCG	LASER	SPTC	DWCG
Profit	12,169	11,905	12,482	10,450	11,482	DNR
Idle	1	14	1	0	0	
Refusal	13	13	13	34	18	
% Empty	16.4	20.7	10.8	19.1	25.0	
% Util.	58.0	54.6	52.9	64.7	79.5	
cpu time	13	3,475	19,887	10	3,809	

Table 7: 80 Task Drayage Problem

PROBLEM	Excess		Constrained	
	LASER	SPTC	LASER	SPTC
50-wide	0.2	1.1	8.1	9.9
50-tight	0.2	0.2	1.9	5.4
50-mixed	0.8	0.7	2.2	5.7
75-wide	0.4	1.3	2.6	6.3
75-tight	0.2	0.2	0.2	3.7
75-mixed	1.0	0.9	7.9	3.1
100-wide	0.5	1.1	9.5	10.4
100-tight	0.0	0.5	0.0	4.3
100-mixed	0.9	0.4	11.2	7.0
61-dray	0.4	2.9	0.7	2.7
73-dray	2.8	2.7	3.9	5.5
80-dray	2.5	4.6	-	-
Average	0.8	1.4	4.4	5.8
Combined Avg	LASER - 2.5		SPTC - 3.5	

Table 8: Objective Function Gap

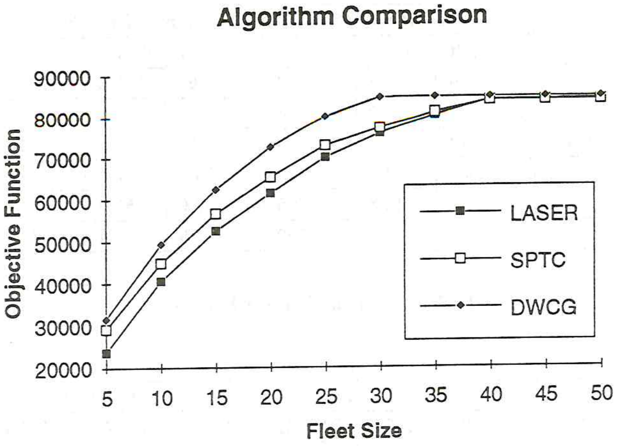


Figure 6: Algorithm Comparison

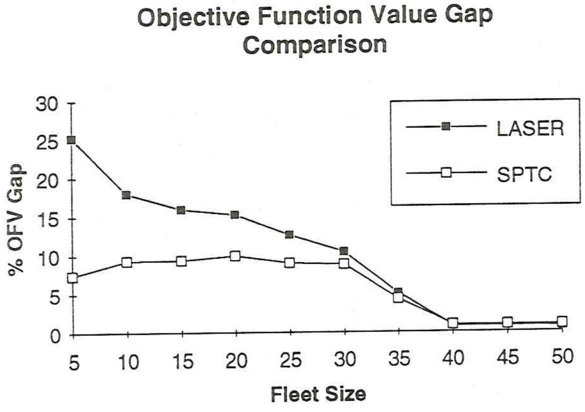


Figure 7: Objective Function Value Gap Comparison

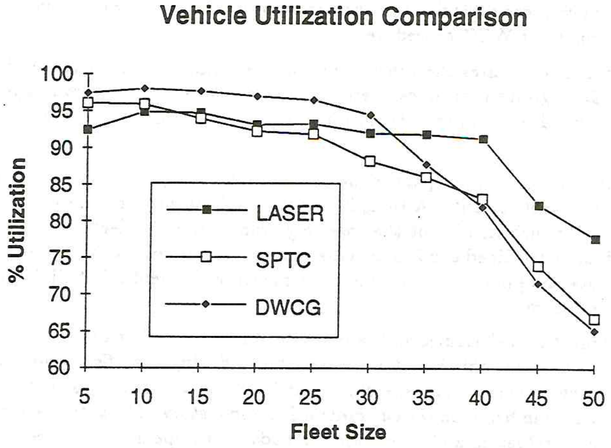


Figure 8: Fleet Utilization Comparison

To evaluate the performance of LASER we conducted a series of numerical experiments on a test problem set that consisted of randomly generated and real-world drayage problems. A Dantzig-Wolfe column generation scheme was designed to provide a lower bound on the optimal solution, and a shortest path based tour construction heuristic was implemented to provide a good upper bound. The LASER algorithm constructs vehicle tours in parallel while the SPTC procedure constructs vehicle tours sequentially, thus we have a breadth-first versus depth-first comparison for DAP algorithms. There are a number of key findings based on the results of the numerical study.

- LASER produces solutions within 1% of LP-optimal solutions for problems with excess fleet capacity.
- LASER solution gap increases by an order of magnitude when the problems are constrained with respect to fleet capacity.
- LASER produces better solutions than SPTC for problems constrained with respect to fleet capacity. LASER and SPTC are comparable for problems with excess fleet capacity.
- SPTC uses fewer vehicles than LASER to cover the same number of loads when there is excess fleet capacity.
- LASER produces better solutions than SPTC for problems with wide and tight time windows. The procedures are comparable for problems with mixed time windows.
- SPTC requires less computational effort than LASER.
- DAP difficulty level is highly dependent on available fleet capacity.

For problems with excess capacity in terms of the number of available vehicles, the LASER algorithm performed well. The LASER algorithm mainly produced results that were within 1% of the DWCG LP-optimal bound. When the fleet size was reduced for the same set of problems the quality of the solutions from LASER deteriorated primarily due to the labelling algorithm's inability to efficiently utilize the available vehicle time. These results were true for both the generated and real-world problems. The fleet capacity constrained DAPs are much more difficult than the excess problems as demonstrated by the large increase in computational effort for the DWCG procedure.

Neither DAP algorithm dominates the other based on the numerical analysis; the SPTC requires less cpu time but the LASER algorithm produces better solutions. The main disadvantage of the depth-first SPTC procedure is its dependence on the order in which the vehicle tours are built when the vehicle fleet is heterogeneous.

The LASER algorithm was clearly incapable of producing high quality solutions when the problem became constrained with respect to fleet capacity. Although in many applications there is excess fleet capacity, there are obviously situations in which this is not the case (e.g. end of month effects). The ability to efficiently solve the DAP under these constrained conditions remains an important unresolved issue and requires further investigation. There exists an opportunity to combine elements of the depth-first SPTC procedure with the breadth-first LASER algorithm.

The dynamic assignment model studied in the paper assumed that all of the relevant problem data was deterministic. In many situations this is not the case, thus reducing the effectiveness of the model. For example, when constructing vehicle tours dispatchers may take into account tasks that have yet to be called in, but probably will be called in based on the customer's demand history. The anticipation of future demands can be addressed through forecasting and stochastic methods which opens up new research opportunities.

References

[1] John J. Bartholdi III and Loren K. Platzman. Heuristics based on spacefilling curves for combinatorial problems in euclidean space. *Management Science*, 34:291–305, 1988.

[2] Martin Desrochers and François Soumis. A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR*, 26:191–212, 1988.

[3] J. Desrosiers, M. Solomon, and F. Soumis. Time constrained routing and scheduling. Technical report, Groupe d'études et de recherche en analyse des décisions, 1992.

[4] Fred Glover, Darwin D. Klingman, Nancy V. Phillips, and Robert F. Schneider. New polynomial shortest path algorithms and their computational attributes. *Management Science*, 31:1106–1128, 1985.

[5] I.J. Lustig. The influence of computer language on computational comparisons: An example from network optimization. *ORSA Journal on Computing*, pages 152–161, 1990.

[6] I.J. Lustig. Simpnet: A package for the network simplex method in c. Technical report, Department of Civil Engineering and Operations Research, Princeton University, 1990.

[7] Warren B. Powell. Optimization models and algorithms: Emerging technology for the motor carrier industry. *IEEE Transactions on Vehicular Technology*, 40:68–80, 1991.

[8] Warren B. Powell and Zhi long Chen. A new algorithm for the shortest path problem wth time windows. Technical report, Department of Civil Engineering and Operations Research, Princeton University, 1994.

A UNIFIED SOLUTION APPROACH TO TIME CONSTRAINED VEHICLE ROUTING AND CREW SCHEDULING PROBLEMS

Jacques Desrosiers, GERAD and Ecole des Hautes Etudes Commerciales de Montreal, Canada
Yvan Dumas, GERAD and Ecole des Hautes Etudes Commerciales de Montreal, Canada
Marius M. Solomon, North-eastern University, Boston, USA
Francois Soumis, GERAD and Ecole Polytechnique de Montreal, Canada

Abstract: The first paper to mention a routing problem with temporal constraints dates back to Dantzig and Fulkerson (1954). Since then, a flurry of activity has been directed at the classical routing model, many of its realistic variants, important generalizations including the temporal aspect, and a myriad of practical applications in vehicle fleet planning and crew scheduling. This research has been reviewed in several insightful surveys written in the early '80s by Magnanti (1981), Bodin, Golden, Assad and Ball (1983) and Carraresi and Gallo (1984).

In terms of solution methodology capable of solving realistic size problems, this field has seen a natural progression from ad-hoc methods to simple heuristics, to optimization-based heuristics and recently optimal algorithms. Helped by continuously better insights into problem structures with time constraints and rapid advances in information systems and computer technology, these optimal methods are now a viable tool for solving practical size problems.

The contribution of this paper is both theoretical and practical in nature. To summarize, we have designed optimal Dantzig-Wolfe decompositions Column Generation schemes embedded in branch-and-bound search trees to solve complex time constrained vehicle routing and crew scheduling problems. This methodology can be extended far beyond the routing and scheduling field. It can be applied to develop optimal approaches to integer programs solved by column generation, an open problem since the early '60s.

From a historical perspective, our research began by analyzing the problem of assigning buses to school trips with flexible departure and arrival times (Desrosiers, Soumis and Desrochers 1984). This problem was formulated as a Multiple Travelling Salesman Problem with Time Windows (M-TSPTW). The model includes a network structure plus additional nonlinear constraints linking flow variables and time variables. Applying a specific Dantzig-Wolfe decomposition to this problem results in a subproblem and a master problem which is the linear relaxation of a set partitioning type problem. The subproblem, generating feasible columns, is a Shortest Path Problem with Time Window constraints (SPPTW) (Desrosiers, Pelletier and Soumis 1983). This research was awarded the first prize for the best contribution from young researchers at the EURO VI congress held in Vienna (1983). This problem was first addressed by Appelgren (1969) for vessel scheduling. In a second paper (Appelgren 1971), the author wrote: "There are fundamental difficulties in combining these integer programming methods with Dantzig-

Wolfe decomposition, since the constraints generated in the master program have to be taken into account in the solution of the subprograms".

Our paper describes an exact column generation scheme integrated in a branch-and-bound search tree, which for the first time is able to optimally solve large problem instances. One reason for the computational and thus the practical success of the approach is that the decomposition keeps most of the difficulties of the original nonlinear time constrained network model in the subproblem structure. The dynamic programming algorithm designed specifically to solve the SPPTW with binary flow requirements closes much of the integrality gap. In this first application, even cuts were correctly added to the master problem while branching decisions were taken on the network flow variables to obtain optimal integer solutions to the original time constrained network-based formulation.

During the past decade, algorithms for several versions of the shortest path problem with time windows have been developed (Desrosiers, Pelletier and Soumis 1983, Desrochers and Soumis 1988 a,b). These algorithms only involve a single cumulative resource on each arc, i.e. the travel time, besides the usual cost component. Since travel time and cost are different components, a list of 2-dimensional labels is kept at each node, from which only Pareto optimal labels are retained. It also was necessary to create constrained shortest path algorithms to handle:
2-resource structures, i.e., time windows and vehicle capacities in vehicle routing problems (Desrochers, Desrosiers and Solomon 1992), additional origin-destination coupling and precedence constraints in pick-up and delivery problems (Dumas, Desrosiers and Soumis, 1991), 2-cycle elimination for non acyclic time constrained networks (Desrochers, Desrosiers and Solomon 1992), multi-resource constrained shortest paths to model complex nonlinear cost functions as well as union and safety rules in bus driver scheduling (Desrochers and Soumis 1989, Desrochers et al. 1992), airline crew scheduling (Desrosiers et al. 1991) and in other applications in the airline context such as monthly crew rostering and aircraft fleet planning with time windows. These algorithms highlight one of the strengths of this research, specifically the efficient implementation of difficult dynamic programming algorithms. In a sense, the more the problems are constrained, the more they are suited to be solved by dynamic programming. This has been successfully exploited in multi-resource constrained shortest path algorithms. Recent applications on very large scale problems, i.e. with more than 3,500 tasks to be scheduled, use up to thirty resources to model very complicated situations.

This paper presents a unified framework for all time constrained vehicle routing, crew scheduling and crew rostering problems. Problems such as the Multiple Depot Vehicle Scheduling Problem (Carpaneto et al. 1989, Ribeiro and Soumis 1991), the Multiple Travelling Salesman with Time Windows, the Vehicle Routing Problems with Time Windows, the Pick-up and Delivery Problem with Time Windows, the Bus Driver Crew Scheduling Problem, the Airline Crew Scheduling Problem, the Airline Rostering Problem, the Airline Fleet Planning Problem, etc., can all be formulated as multi-commodity network flow problems with additional resource constraints. This model easily accounts for multiple depots, or multiple vehicle and crew types, and even initial conditions for each specific vehicle or crew member. It therefore can be used as an optimization tool in a planning phase as well as a reoptimization tool in an operational mode.

Applying Dantzig-Wolfe decomposition Column Generation scheme to this unified model which possesses a block angular structure results in a set partitioning type problem as the master, and specialized shortest path problems with resource constraints as subproblems. The forward dynamic programming algorithm used in the solution of the shortest path problems can handle complex nonlinear cost functions (in terms of flow and resource variables), while the resource definition and the network design can deal with time windows and vehicle capacities, coupling and precedence constraints, local constraints such as union and security rules, etc.

Global constraints, such as the number of available aircraft of each type or the number of part time and full time workers, are left in the master structure.

The solution of the master problem which is a linear program, provides a lower bound on the original multi-commodity network flow model. The integrality gap is very small in practice. This is the only practical method which can give such a lower bound and thus an estimate of the quality of a feasible solution.

Cuts and branch-and-bound decisions are taken on the multi-commodity flow model rather than on the derived set partitioning type (the master) problem. Decisions using flow and resource variables are thereafter transferred to the adequate structure.

Reapplying the decomposition process to the modified multi-commodity model, branching decisions and cuts appear either in the master problem (e.g. cuts on the minimum number of aircraft used) or in the subproblem structures (e.g. 0-1 branching decisions on flow variables or time window divisions).

This multi-commodity flow model / column generation / branch-and-bound scheme implicitly (mathematically) manages trillions of feasible columns. In practice, several thousands columns are sufficient. The basic principles of Dantzig-Wolfe decomposition/Column Generation scheme, incorporated in a branch-and-bound search tree to achieve integrality, have been around for decades. And yet, for many years, the field has concluded that "Dantzig-Wolfe decomposition does not work", a feeling that continues to persist among many members of the Operations Research community. Our research team has demonstrated that the difference between a basic principle and a practical tool, as with any complex technological problems, is thousands of hours of development, testing and refinement. The result inevitably is a much deeper understanding of the method and its application to a particular broad class of problems.

In the last few years, commercial software systems based on this optimization methodology have been developed. Their use in Montreal, Lyon, Paris, Toulouse, Tokyo, etc., has resulted in substantial savings for the companies involved. For example, the application of this methodology to Air France crew scheduling problems has resulted in savings of over 6% which amounts to tens of millions of dollars per year. In contrast with many commercial applications, this research demonstrates that the application of operations research methods can be undertaken without compromising mathematical rigor.

Finally, the paper highlights our most recent computational results in a number of time constrained routing and scheduling environments.

References

APPELGREN L.H. (1969), A Column Generation Algorithm for a Ship Scheduling Problem, Transportation Science 3, 53-68.

APPELGREN L.H. (1971), Integer Programming Methods for a Vessel Scheduling Problem, Transportation Science 5, 64-78.

BODIN L., GOLDEN B., ASSAD A. and BALL M. (1983), Routing and Scheduling of Vehicles and Crews: The State of the Art, Computers and Operations Research 10, 62-212.

CARPANETO D., DELL'AMICO M., FISCHETTI M. and TOTH P. (1989), A Branch and Bound Algorithm for the Multiple Vehicle Scheduling Problem, Networks 19, 531-548.

CARRARESI P. and GALLO G. (1984), Network Models for Vehicle and Crew Scheduling, European Journal of Operational Research 16, 139-151.

DANTZIG G. and FULKERSON D. (1954), Minimizing the Number of Tankers to Meet a Fixed Schedule, Naval Research Logistics Quarterly 1, 217-222.

DESROCHERS M., DESROSIERS J. and SOLOMON M.M. (1992), A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows, Operations Research 40, 342-354.

DESROSIERS J., DUMAS Y., DESROCHERS M., SOUMIS F., SANSO B. and TRUDEAU P. (1991), A Breakthrough in Airline Crew Scheduling, Proceedings of the 26th Annual Meeting of the Canadian Transportation Research Forum, Quebec City, May 28-31, 464-478.

DESROCHERS M., GILBERT J., SAUVE M. and SOUMIS F. (1992), CREW-OPT: Subproblem Modelling in a Column Generation Approach to Urban Crew Scheduling, Computer-Aided Transit Scheduling, (M. Desrochers and J.M. Rousseau, eds.), Lecture Notes in Economics and Mathematical Systems 386, Springer Verlag, Berlin Heidelberg, 395-406.

DESROCHERS M. and SOUMIS F. (1988a), A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows, INFOR 26, 1991-212.

DESROCHERS M. and SOUMIS F. (1988b), A Reoptimization Algorithm for the Shortest Path Problem with Time Windows, European Journal of Operational Research 35, 242-254.

DESROCHERS M. and SOUMIS F. (1989), A Column Generation Approach to the Urban Transit Crew Scheduling Problem, Transportation Science 23, 1-13.

DESROSIERS J., PELLETIER P. and SOUMIS F. (1983), Plus Court Chemin avec Contraintes d'Horaires, RAIRO 17, 357-377.

DESROSIERS J., SOUMIS F. and DESROCHERS M. (1984), Routing with Time Windows by Column Generation, Networks 14, 545-565.

DUMAS Y., DESROSIERS J. and SOUMIS F. (1991), The Pick-up and Delivery Problem with Time Windows, European Journal of Operational Research 54, 7-22.

MAGNANTI T. (1981), Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects, Networks 11, 179-214.

RIBEIRO C.C. and SOUMIS F. (1991), A Column Generation Approach to the Multiple Depot Vehicle Scheduling Problem, Operations Research (to appear).

Dynamic traffic assignment in congested networks : an iterative algorithm.

Luce Brotcorne Daniel De Wolf Martine Labbé
Service de Mathématiques de la Gestion
Université libre de Bruxelles CP 210/01
Boulevard du Triomphe B-1050 Bruxelles, Belgium

Abstract

1 Introduction

Congestion has been growing dramatically over the past 20 years, leading to gridlock conditions in many European cities. Moreover, all forecasts asses that without any change, the urban transportation network will be totally congested by the year 2000. It is clear that we can no longer build our way out of the economic, political and social constraints involved with the congestion problem particularly in urban areas.

In this context, the modeling of commuter behavior can be used to help public authorities to test the impact of various transportation policies such as the enlargement of some bottlenecks, the variation of flexible work start times, the addition of parking at some entering points, ... etc.

Previously, most network equilibrium models focused on the static description of traffic flows, implying that flows and travel times are invariant over the duration of the peak period. Under this assumption, users respond to congestion only through the route choice decision thus, the departure time decision is ignored. However, the departure time choice influences traffic congestion. Indeed, commuters may leave their origin early in order to avoid congestion but then experience a long wait at their destination, or they may reduce their schedule delay but then experience a large travel time.

In answer to the inadequate assumption of static flows over the peak period, a number of studies have suggested *dynamic* traffic assignment models. But, the presence of time-dependent congestion makes the dynamic equilibrium problem much more difficult than the static one. So, in most existing models, major simplifications are made to make the problem more tractable. Moreover, since the proposed dynamic network equilibrium models are usually non convex, the algorithms used to solve the problem are heuristic.

One of the first studies on this topic was produced in 1978 by Merchant and Nemhauser [3, 4]. Assuming departure times exogenously given, they compute a dynamic social optimum (DSO) on a single destination network. More precisely, they determine arc flows in order to minimize the total travel time endured by the commuters on the network. The model is nonlinear and nonconvex. Then, with a restrictive assumption on the arc traversal time functions depending on arc flows, the problem is solved by piecewise linear approximation and use of the Simplex algorithm. Other models require simplified networks (for example a bottleneck, or a single destination network, ...), or they only determine the paths of the users and suppose that departure times are exogenously given (despite the fact that they are important decisions variables). Finally some other models [5] transform the dynamic problem in a static one on an extended network. But these models contain a great number of variables, and it is not sure that the user equilibrium computed on the extended network will be a dynamic user equilibrium.

2 A Dynamic User Equilibrium Model

We consider a dynamic model for traffic assignment in urban transportation networks. Time is considered as a discrete variable. The transportation demand is given by the number of people travelling from the same origin to the same destination and having the same desired arrival time.

The model determines both path and departure time of commuters in order to reach a user equilibrium. More precisely, an affectation of people to choices of path and departure time is said to be in a user equilibrium situation if no user can increase his satisfaction by unilaterally changing his departure time or his path. So, we assume that commuters try to minimize their own disutility function which integrates constant link costs, the path travel time and a penalty for arrival at destination earlier or later than a given desired time.

The arc traversal time function depending on arc flow is defined in order to satisfy the first in first out assumption. More presisely, let g_a [1] a nondecreasing function defined on IR^+ with values in the set

[1] The function g_a correspond to the classical "arc performance" function that links the travel time with the flow present on the arc a at time t noted $f_a(t)$. A commonly used "arc performance function" is the function developped by the Bureau of Public Roads (1964), named the B.P.R fuction.

of discrete times. The traversal time of the arc a for a user entering at time t, noted $tt_a(t)$, is defined as the maximum between the two following terms: the function g_a evaluated at the flow present on the arc a at time t noted $f_a(t)$, and the travel time experienced by a user who entered arc a in the preceding time period. Formally:

$$\begin{aligned} tt_a(0) &= g_a(0) \\ tt_a(t) &= \max\left\{g_a\left[f_a(t)\right], tt_a(t-1)-1\right\}, t = 1, \ldots T \end{aligned}$$

The first in first out property is a natural assumption in urban congested network since generally vehicles cannot overtake each other. However this assumption would not be satisfied if the travel time function were simply defined as $tt_a(t) = g_a(f_a(t))$. Indeed, if at time t a great number of users leave the arc a, its traversal time will sharply decrease. So, it would be possible that users entering the arc at time t would leave it before those who entered during a preceding time period.

3 Algorithm

The **solution algorithm** we propose is a dynamic generalization of the Frank-Wolfe algorithm applied to the static user equilibrium model. It can be decomposed into four main steps.

- **Initializations:** According to the free flow arc traversal times, an optimal choice of path and departure time is computed for each set of people travelling from the same origin to the same destination and having the same desired arrival time. Then, all set of people are assigned to the corresponding choice .

- **The loading**: Given an allocation of users to the different choices of path and departure time, this step determines the flows present on each arc and the corresponding arc traversal times. This is done chronologically: we determine the first set of people leaving an origin node in the morning and we load this set on the first arc of their path. Then, we look for the next set of people leaving a node (an origin, or an intermediaire node) until all people have reached their destination..
- **Computation of an optimal choice of path and departure time.** According to the arc traversal times computed in the loading step, a new optimal choice of path and departure time is computed for each set of users travelling from the same origin to the same destination and having the same desired arrival time.
- **convergence test**: for each set of people travelling from the same origin to the same destination and having the same desired arrival time, if the disutility function value corresponding to the new choice of path and departure time (which is in fact the minimal disutility function value corresponding to the associated set of people) is equal to those of all the used choices, then the algorithm stops. Otherwise it reallocates people among the old and the new choices.

The main problem of our algorithm is the computation of an optimal choice of path and departure time. This procedure relies upon decomposition into two subproblems. First, for a given departure time, finding the best path for a given customer is a NP-Hard problem.

However, assuming that there is no constant link cost, this problem can be solved by applying Dijkstra'salgorithm [2]. Then, computing the optimum departure time reduces to a global optimization problem. This is solved by adapting Piaviskii's algorithm for the minimization of univariate lipschitz functions to the case of functions which are one-sided lipschitz only [1]. Note that this main problem of determining the optimal path and departure time is much simpler when there is no constant link costs. Indeed, in this case, the best path can be determined by applying Dijkstra's algorithm.

Finally, we present the results of the algorithm on a simple example for which optimality conditions can be easily verified.

References

[1] DE PALMA A.and P. HANSEN, Optimum departure times for commuters in congested networks, *Northwestern University and Rutcor, Rutgers University*, 1990.

[2] DE PALMA A., HANSEN P. and LABBE M., Commuters paths with penalties for early or later arrival time, *Transportation science 24, No 4*, pp 276-286, 1990

[3] MERCHANT D.K and NEMHAUSER G.L., A model and an algorithm for the dynamic traffic assignment problem, *Transportation Science, 12*, pp 183-199, 1978.

[4] MERCHANT D.K and NEMHAUSER G.L., Optimality conditions for a dynamic traffic assignment model, *Transportation Science, 12*, pp 200-207, 1978.

[5] DRISSI-KAITOUNI and HAMEDA-BENCHEKROUN, A Dynamic Traffic Assignment Model and a Solution Algorithm, *Transportation Science*, **26, 2**, pp 119-128, 1992.

A System Optimal Dynamic Traffic Assignment Model with Distributed Parameters

E. Codina, J. Barceló
Departament d' Estadística i Investigació Operativa
Facultat d' Informàtica, Pau Gargallo, 5. 08028 Barcelona (Spain)
Universitat Politècnica de Catalunya, Spain
Tel. +34.3.401 7033, Fax: +34.3.401 7040

June, 1994

1 Abstract

In this paper a multidestination system optimal dynamic traffic assignment model with distributed parameteres is examined. This model can be considered an extension of previous and well known models such as those of Merchant and Nemhauser (1978) and Friesz (1990). Flow dynamics of the model is based on an extension of the simple continuum model for flows composed by several commodities with equal propagation characteristics. An important property of flows following this model is that no overtaking can occur between flows of different commodities. Several approximations by means of ODE systems to the extended continuum model are discussed in terms of their stability properties. In the case of flow dynamics of the type constant propagation speed and vertical queues at links it is possible the *stable* approximation of the proposed dynamic system optimal model by means of optimal control problems. For one of the possible stable approximations it is shown how a strengthened Courant-Friedrichs-Levy condition ensures no overtaking and FIFO observance at vertical queues of the approximating optima control problem. Finally, and for the case of a single destination in the network, the application of an extremals calculation method for optimal control problems developed by the authors in previous papers is also shown and favorable conditions for its application are discussed.

2 Introduction

In the last years dynamic behaviour of urban traffic flows under route guidance systems has been modeled by some authors using a deterministic optimal control approach formulation. This formulation has emerged as a dynamic extension of wardropian equilibrium principles widely used in urban transportation planning. Deterministic optimal control based formulation was stated initially by J.F. Luque and T.L. Friesz [14] (1980) coming from a Merchant and Nemhauser model [15], [16] for networks with a single destination. M. Carey [4], [5] analyzed the constraints qualification of Merchant and Nemhauser's model. Also models with multiple destinations have been proposed by W. Wie, Y.L. Friesz and R.L. Tobin [23]. All these models had the common characteristic of using the so called *exit link functions* in order to model traffic dynamics. Models without exit link functions also based on deterministic optimal control have been proposed by B. Ran, D. Boyce and Leblanc L.J. [19]. Although models with and without exit link functions provided a compact formulation they presented some non adequate properties for the evolution of traffic flows. Thus, models without exit link functions presented unproper characteristics when applied to traffic flows and in [9] E. Codina shows paradoxes of the solutions given by dynamic traffic assignment models without exit link functions that lead to an extension of the Dafermos-Sparrow theorem for static traffic assignment models. Instantaneous propagation characteristics of flow dynamics in models with exit link functions are analyzed in [8] and it is shown how this unproper characteristic vanishes as a simple continuum model is approximated. Another unproper characteristic for multidestination models already shown by M. Carey in [5] consists of the conflict arising when flows with different destinations exit from links and the observation of FIFO queueing disciplines by these flows. However, as suggested by Papageorgiou in [18], models using exit link functions can be viewed as an extension of a model with distributed parameters (E. Codina [9]).

This paper presents a general continuous System Optimal Dynamic Traffic Assignment model for multidestination networks. Flows of this model verify the inviscid PDE for stationary and irrotational flows and this ensures that neither instantaneous propagation nor overtaking may occur between flows of different commodities at links. Also, because of its basic trafic behaviour characteristics, i.e. finite propagation speed, wave dispersion and effects of congestion (i.e. speed reduction and spillback) this model can be taken as a reference. In fact, it is shown how Merchant and Nemhauser's model can be considered a discrete approximation to the continuous system optimal model presented here for the case of density dependent propagation speed and how Friesz's and Wie's models are also approximations by means of continuous time optimal control in the case of constant propagation speed on links. Third section of this paper gives a brief description of Merchant and Nemhauser's model and Carey's work and the problem of overtaking between flows of different commodities at links. Section 4 starts with a basic description of the simple continuum model and discusses *unstability* characteristics of their first order regressive difference approximations, i.e. Euler explicit method and implicit method and the need of vertical queues at the exit of links when this approximations are used. Next an extension to the multicommodity case of the simple continuum model on a link is presented. The no overtaking of trajectories characteristic of this extended model is remarked and it is shown that, when constant propagation speed and vertical queues at the end of links hold, a first order implicit regressive finite difference approximation gives solution flows with no overtaking at links and FIFO observance at vertical queues, provided that a strengthened Courant-Friedrichs-Levy condition is observed by the discretization mesh. This characteristics can be expressed by a set of *linear* inequalities, thus eliminating nonconvexities of earlier formulations that expressed FIFO observance. In section 5, several approximations by means of ordinary differential equations (ODE) to the simple continuum model for the single commodity case are reviewed and the known fact that nonlinearities (i.e. non constant flow-density relationship) imply *unstability* of first order solution methods for the ODE systems is outlined. Section 6 presents the continuous system optimal dynamic traffic assignment model of this paper. Because of unstabilities due to nonlinearities a continuous time optimal control approximation is presented, only for the case of constant propagation speed and vertical queues at the end of links. In order to guarantee no overtaking and no destination selective output form vertical queues at links, results obtained in section 4 regarding elimination of nonconvexities are used and a nonlinear optimization problem is formulated to solve the optimal control approximation. It must be remarked that constraints of this nonlinear optimization problem are now *linear*. Finally and for the case of a single destination network, conditions that permit the application of an extremal calculation method for optimal control problems developed in [9] and that decompose the previous nonlinear programming formulation are briefly discussed.

3 Merchant and Nemhauser's Model. Preliminaries

In 1978 Deepak K. Merchant and George L. Nemhauser (refs. [15] y [16]) published two papers describing a system optimal dynamic traffic assignment model formulated as a nonlinear optimization problem for networks with a single destination. Because of the nonconvexity of their model Merchant and Nemhauser could not analyze constraints qualification and uniqueness aspects. This was done by M.Carey ([4] , [5]) by means of a modified model. This model is also a precursory one of "continuous time" models presented by Friesz ([10]), B.W.Wie ([23]). Also an optimal control distributed parameters model presented in this paper and suggested in ([9]) can be considered as an extension of Merchant and Nemhauser's model. Because of their historical relevance we shall give a summarized description of Merchant and Nemhauser and M.Carey's models as well as of their properties.

Merchant and Nemhauser's System Optimal Model. Let a network be represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ consisting of a set of nodes $\mathcal{N}$ and a set of links $\mathcal{A}$ with traffic flows moving towards a single destination $q \in \mathcal{N}$ in the graph. Let us denote by $\mathcal{I}(i)$ and by $\mathcal{E}(i)$ the set of links entering and emerging from node $i \in \mathcal{N}$ respectively. Let's supose a finite horizon of time $[0, T]$ divided in N time subintervals of equal length $\delta = T/N$ and let us consider the following magnitudes:

$x_{a,j}$ Number of vehicles in link $a \in \mathcal{A}$ (or the average density of vehicles) and

	time subinterval j.
$u_{a,j}$	Number of incoming vehicles at link $a \in \mathcal{A}$ and j-th time subinterval.
$v_{a,j}$	Number of outgoing vehicles from link $a \in \mathcal{A}$ and j-th time subinterval.
$p_{i,j}$	The number of vehicles entering the network at node $i \in \mathcal{N}$ in j-th time subinterval with destination q.
$X_{a,j}(\cdot)$	Cost functions for link $a \in \mathcal{A}$ and j-th time subinterval.

Exit link flows are assumed to verify the following hypothesis:

1. $v_a = w_a(x_a)$, i.e. exit flow from a link depends on the number of vehicles at j-th time subinterval on the link $a \in \mathcal{A}$. Merchant and Nemhauser refer to $w_a(\cdot)$ as "exit link functions" (ELF).

2. Functions $w_a = w_a(x_a)$ are non decreasing, continuous and concave and verify $0 \leq w_a(x_a) \leq x_a$.

3. Functions $w_a(\cdot)$ also verify a *saturation* property: $lim_{\,\omega \to \infty} \; \frac{dw_a(\omega)}{d\omega} \; = 0$

With all these hypothesis, Merchant and Nemhauser model can be reformulated as the following optimization problem:

M-N 1 (Reformulation of Merchant-Nemhauser's model **M-N 1** using exit link functions as side constraints.)

$$
\begin{array}{lll}
Min_{\,x_j, u_j, v_j, s_j} & \sum_{j=1}^{N} \sum_{a \in A} X_{a,j}(x_{a,j}) & (L.M.) \\
s.t. & x_{j-1} - x_j - v_j + u_j = 0 & \lambda_j \\
 & B_+ u_j - B_- v_j + e_q s_j = p_j & \alpha_j \\
 & v_{a,j} = w_a(x_{a,j-1}) & \vartheta_{a,j}, \;\; \forall\, a \in \mathcal{A}, \;\; j = 1, \ldots, N \\
 & u_j \geq 0, \;\; v_j \geq 0, \;\; x_j \geq 0, \;\; s_j \geq 0 & \varsigma_j, \; \xi_j, \; \bar{\eta}_j, \; \varepsilon_j \\
 & (\; x_0 \; known \;) &
\end{array}
$$

(Although Merchant and Nemhauser did not prove that the problem verifies a constraint qualification we write on the right hand of each constraint its corresponding Lagrange multiplier (L.M.))

The first group of constraints is a vector notation for $x_{a,j-1} - x_{a,j} - w_a(x_{a,j-1}) + u_{a,j} = 0, \; a \in \mathcal{A}$, and can be viewed as state equations. The second group of constraints, $B_+ u_j - B_- w(x_j) + e_q s_j = p_j$, are balance equations for flows at nodes being B_+ and B_- matrices derived from the incidence matrix B of graph $\mathcal{G}$ with $(B_+)_{i,j} = 1$ if $(B)_{i,j} = 1$ and zero otherwise $(B_-)_{i,j} = 1$ if $(B)_{i,j} = -1$ and zero otherwise and s_j arrival flow at destination q in time subtinterval j-th. It must be pointed out that because of $0 \leq w_a(x_a) \leq x_a$ and $u_{a,j} \geq 0$ nonnegativity constraints on variables x_j are *redundant*. In this formulation exit flow variables $v_{a,j}$ are used explicitly and side constraints are derived from the hypothesis of exit flow at links thus suggesting a network structure. We shall call "time-expanded networks" to networks originated by the linear constraints in problem **M-N 1** and we shall refer to $\mathcal{G}$ the as "original network".

Because of analytical difficulties of Merchant and Nemhauser's model, M. Carey considered in [4] and in [5] a similar model that can be formulated as **M-N 1** "almost equal". The only difference is that exit flows on links are *limited* by the exit link functions: $0 \leq v_{a,j} \leq w_a(x_{a,j}) \quad (\leq x_{a,j}\,)$. It is worth noting that redundancy of nonnegativity constraints for state variables $x_{a,j}$ remains. Carey denominates the slack $\sigma_{a,j} = w(x_{a,j}) - v_{a,j}$ as "control" for the link a and provides in [5] two sets of sufficient conditions on the objective costs (**CFC** conditions) and the exit link functions (**EFC** conditions) in order to solutions of the modified Merchant Nemhauser's model **M-N 1** provide *null* controls $\sigma_{a,j}$. For this modified model M. Carey proved that the modified model satisfies Kuhn Tucker conditions. It must be noted that Model **M-N 1** and also the modified model due to Carey is restricted to a single destination in the network and also that **CFC** conditions on cost functions may not hold in many practical cases.

For the extension to multiple destinations these models present a nondessirable property when their solutions are compared to traffic flows behavior. This nondesirable property is commonly known as "FIFO discipline violation". On a road network, traffic of different types entering the same link at approximately the same time t will pass each other but their speeds will not differ too much and therefore, approximately equal travel times to traverse links for flows with different destinations in the network must be expected. Consequently it will appear as "anomalous" any solution for which flows entering the link at time $t' > t$, (after cars entering at time t) leave the link at time τ' *before* cars that entered at time t, thus reflecting that an overtaking occurs between cars of different commodities. The practical implications of the "FIFO discipline violation" in a multidestination situation are of great relevance. If solutions of the model are to be applied on *real* traffic networks (i.e. for route guidance purposes), then, as imposed by "controls" $\sigma^q_{a,j}$, $q \in D$, a *selective* exit of cars from a link would be required accordingly to the different car destinations (some cars with destination q would have to wait or travel at a lower speed in order to allow cars with destination q' exit the link) and this strategy seems impossible to implement in practice or, at least, adds huge difficulties in real traffic networks, where the natural magnitudes that can be controlled are the aggregated exit flow from link a, $v_a = \sum_{q \in D} v^q_a$ or input flow, $u_a = \sum_{q \in D} u^q_a$. As stated in [5], constraints of the type $w_a(\sum_{q \in D} x^q_{a,j}) \geq \sum_{q \in D} v^q_{a,j}, 1 \leq j \leq N$, allow any order in exiting a link for the commodities and only ensure that exit flow capacities $w_a(\cdot)$ at links are satisfied. In [6] Carey formulates four classes of constraints that ensure the FIFO discipline of multicommodity flows or are consequence of the observation of the FIFO discipline. They all require that problems be formulated with flow variables $h_{a,k,\bar{k}}$ entering a link at time subinterval k and leaving the link at time subinterval $\bar{k} \geq k$. The first of these classes of constraints needs additionally the use of binary decision variables. Each of the four classes of constraints result in a nonconvex constraint set and results in nonlinear-integer programming formulations.

Empirically, Carey suggests in [6] to solve dynamic assignment problems without constraints for FIFO observance, analyze the degree of overtaking or FIFO violations, and introduce FIFO constraints if necessary. Also through empirical work, Carey intuits that a certain degree of correlation exists between sharp flow fluctuations and FIFO discipline violation and thus, the technique previously described is specially suggested for the case of traffic flows, because their *smooth fluctuations in time* result in the observation of FIFO discipline or in its approximate verification.

4 Considerations on State Equations in Dynamic Models

State equations in Merchant and Nemhauser's model can be considered an explicit first order approximation of the well known "simple continuum model" based on inviscid flow equation (see, for instance [2]) with the role of exit link functions being parallel to the role of flow-density relationship in the simple continuum model. In this section, after an exposition of the basics on the simple continuum model, we analyze properties and inconvenients of its approximnation by explicit and implicit first order regressive-differences.

We show in this section that implicit approximations, when propagation speed is constant, present the desirable property that solution flows present only a reduced degree of overtaking that vanishes as the discretization mesh increases and a strengthened Courant-Friedrichs-Levy (CFL) condition is observed by the discretization mesh (see, for instance reference [17]) and that this is a consequence of convergence[1] properties of the approximation method to the exact solutions of the model. On the other hand we outline the well known handicap of the unstability of the first order approximations in presence of nonlinearities (non constant propagation speed) and congestion.

Let us consider a link as an unidimensional continuum. Let $x(z,t)$ be the vehicle density of the link and $\jmath(z,t)$ be the flow on the link at time t in position z, respectively, and let speed ω be assumed to depend directly on density following the the flow-density relationship $\jmath = x\,\omega(x)$. The simple continuum model can be stated as:

$$\left.\begin{aligned} \frac{\partial x}{\partial t} + \frac{\partial \jmath}{\partial z} &= 0 \\ \jmath(z,t) &= x(z,t)\,\omega(x(z,t)) \end{aligned}\right\} \Rightarrow \quad \frac{\partial x}{\partial t} + \varphi(x)\,\frac{\partial x}{\partial z} = 0 \qquad (\varphi(x) = \omega(x) + x\,\frac{d\omega}{dx}\,) \qquad (1)$$

[1] **Implicit approximations, when propagation speed is constant, are *numerically stable*. As the approximation method is *consistent* then, by Lax's equivalence theorem (see, for instance [2]) then, there exists *convergence* of the solutions of the implicit approximation to the exact solutions of the PDE as the grid is refined.**

We shall assume a maximum capacity on this link $\hat{x}$ and we shall consider two classes of speed functions $\omega(x)$. The first class are simply positive constant functions. The second class of functions verify the following conditions:

1. $\omega(x)$ is at least once differentiable on $[0, \hat{x}]$ and is decreasing on $[0, \hat{x}]$ with $\omega(0) > 0$ and $\omega(\hat{x}) = 0$.
2. Flow function $\jmath(x) \triangleq x \cdot \omega(x)$ has a single stationary point (maximum) at density $\bar{x} \in]0, \hat{x}[$. We shall refer to $\bar{x}$ as *critical density*. We shall use the term "congestion" when densities are greater than the critical density $\bar{x}$.

Boundary conditions for PDE 1 Boundary conditions that we will consider in order to determine a unique solution of the previous PDE on $[0, \hat{z}] \times [0, T]$ are:

1. Initial density distribution on $[0, \hat{z}]$, $x(z, 0) = h(z)$
2. Input flows to the link at each time t: $\jmath(0, t) = u(t)$
3. *Either* exit flows from the link at each time t: $\jmath(\hat{z}, t) = v(t)$ *or*
4. Use an additional variable κ, the total number of accumulated vehicles at the end of the link, ruled by the differential equation $\dot{\kappa} = x(\hat{z}, t)\omega(x(\hat{z}, t)) - v(t)$.

Together with conditions 1 and 2, condition 3 can not determine a solution in the case of a positive constant speed-density relationship. However it will be possible to determine a solution if boundary condition 3 is substituted by 4. Conditions 1, 2 and 4 are equivalent to model a *vertical queue* at the end of the link. Boundary conditions at $z = \hat{z}$ are related clearly with the effects of a throuput capacity reduction at the link exit (i.e. because of reduction in the number of lanes in the next link) and it must be outlined that when it is possible to impose boundary conditions of the type 3 it is then possible to model spillback[2].

Having in mind potential theory for bidimensional stationary and irrotational flows in fluid mechanics (see, for instance[21]), density $x(z, t)$, solution of $x_t + \varphi(x)x_z = 0$, and its associated flow $\jmath(z, t) = x(z, t) \cdot \omega(x(z, t))$, are the components of a vector field $\vec{E} = (-x, \jmath)$ orthogonal to flow trajectories, and $\vec{E}$ comes from an at least twice differentiable potential function $U(z, t)$, ($-x = \frac{\partial U}{\partial z}$, $\jmath = \frac{\partial U}{\partial t}$), at regions where functions x, $\jmath$ are differentiable. On the other hand conservation of flow ensures continuity of potential function $U(z, t)$ at points where continuity of density function $x(z, t)$ fails. Thus, even if functions $u(t)$ or $v(t)$ determining boundary conditions for $x_t + \varphi(x)x_z = 0$ have a finite set of discontinuity points, flow trajectories shall *never intersect* and a strict observation of the FIFO principle must hold.

Finite difference methods to solve PDE (1) rely on the approximation made to the term $\frac{\partial \jmath}{\partial z}$. If rectangle $[0, L] \times [0, T]$ is discretized in a grid mesh of size (δ_z, δ_t), then approximations to the solution of (1) are made at points (z_k, t_i), $k = 0, 1, 2, \ldots, M = L/\delta_z$ and $i = 0, 1, 2, \ldots, N = T/\delta_t$. A first order regressive approximation to the term $\frac{\partial \jmath}{\partial z}$ leads to explicit or implicit methods:

$$x_k^{i+1} = x_k^i - \nu(\jmath_k^i - \jmath_{k-1}^i) = \nu \jmath_{k-1}^i + x_k^i(1 - \nu\omega(x_k^i)) \quad (\textit{Euler explicit})$$

$$x_k^{i+1} = x_k^i - \nu(\jmath_k^{i+1} - \jmath_{k-1}^{i+1}) \Rightarrow x_k^{i+1}(1 + \nu\omega(x_k^{i+1})) = x_k^i + \nu \jmath_{k-1}^{i+1} \quad (\textit{implicit})$$

$$(\textit{ as } \jmath_k^i = x_k^i \omega(x_k^i), \quad (\nu = \frac{\delta_t}{\delta_z}))$$

It must be noted that methods based on regressive approximations of the term $\frac{\partial \jmath}{\partial z}$ *do not* allow the inclusion of boundary conditions at $z = \hat{z}$ of the type 3, i.e. exit flow $v(t)$ can not be imposed at $z = \hat{z}$ and can only reproduce boundary conditions of the type 4. Thus the use of regressive approximations

[2] It is worth noting at this point that modifications made by Carey to Merchant and Nemhauser model is able to reproduce horizontal queues as exit flows on links are subject to inequalities $\jmath_k^i \leq x_k^i \omega(x_k^i)$ instead of equalities. However controls or "slacks" $\sigma_k^i = x_k^i \omega(x_k^i) - \jmath_k^i$ are guaranteed to be nonnull only under very restrictive assumptions on the cost functions.

to the term $\frac{\partial \jmath}{\partial z}$ allows only to model vertical queues at the end of the link by considering an extra relationship expressing total number of vehicles κ^i at period i-th as, for instance:

$$\kappa^{i+1} = \kappa^i + \delta_t(x^i_M \omega(x^i_M) - v^i)$$

Explicit methods for PDE (1) require the observance of CFL condition (see, for instance [17]). This condition requires that discretization mesh (δ_z, δ_t) observes the inequality $\hat{\omega}\frac{\delta_t}{\delta_z} \leq 1$, with $\hat{\omega} = \omega(0)$ the maximum speed. This condition ensures two desirable properties for Euler explicit method:

1. Flows do not propagate at speeds higher than the maximum speed $\hat{\omega}$.
2. If densities at $t = 0$ and input flows at $z = 0$ are nonnegative then solutions x^i_k are also nonnegative.

Unfortunately it is known that Euler explicit method is *unstable even* if CFL condition is observed (see [2]) no matter the class of function for the speed-density relationship. We can consider a link subdivided into M sublinks and identify $\nu x \omega(x)$ with an exit link function and the term $\nu \jmath^i_k$ with inputs u during time period δ_t in Merchant and Nemhauser's model. Thus, as Merchant and Nemhauser's model uses Euler explicit method to approximate PDE (1) it is subject to *numerical unstabilities* even for time discretization length $\delta_t \longrightarrow 0$.

Previous results regarding FIFO properties can be shown to hold in a countinuous model when flow on a link is composed by a set of commodities $q \in D$, being thus possible to enunciate lemma 1 below. For brevity we enunciate it here without proof.

LEMMA 1 (*FIFO observation for trajectories of the system* $x^q_t + (x^q \cdot \omega(x))_z = 0,\ q \in D$)

Let $x^{*q}(z,t)$, $q \in D$, be solutions of the PDE system:

$$\frac{\partial x^q}{\partial t} + \frac{\partial(x^q \cdot \omega(x))}{\partial z} = 0,\ \ q \in D,\ \ (x = \sum_{q \in D} x^q) \tag{2}$$

fixed by initial conditions $x^q(0,t) = u^q(t)$, $x^q(\hat{z},t) = v^q(t)$, $\forall t \in [0,T]$ and $x^q(z,0) = h^q(z)$, $\forall z \in [0,L]$, $q \in D$, with $u^q(t)$, $u^q(t)$ and $h^q(z)$ functions with discontinuities in a finite set of points of their respective intervals. Then:

1. *Trajectories in $[0,L] \times [0,T]$ are common to all commodity flows and are given by $\dot{z} = \omega(x^*(z,t))$ and no intersection occurs between two trajectories,* even *at points where densities $x^{*q}(z,t)$ may be discontinuous.*
2. *If a point (z,t) is a discontinuity point for density $x^{*q}(z,t)$ then it is a discontinuity point for all other densities $x^{*q'}(z,t)$, $q' \in D$. Even more, discontinuity points for densities, if they exist, verify differential equation:*

$$\frac{d\rho}{d\tau} = \frac{\jmath^q_+ - \jmath^q_-}{x^q_+ - x^q_-}\Big|_{(\rho,\tau)} \ \left(= \frac{\jmath^{q'}_+ - \jmath^{q'}_-}{x^{q'}_+ - x^{q'}_-}\Big|_{(\rho,\tau)},\ \forall q' \in D\right) \tag{3}$$

with $\jmath^q_+$, x^q_+, $\jmath^q_-$, x^q_- being flows and densities at both sides of the curve $\rho(\tau)$. ($(\rho(\tau),\tau)$ can be considered a "collision" point of two flow waves).

3. *Composition ratios $r^q(z,t)$ given by:*

$$r^q(z,t) = \frac{\jmath^q(z,t)}{\jmath(z,t)} = \frac{x^q(z,t)}{x(z,t)} \ \ if\ x(z,t) > 0,\ \ r^q(z,t) = 0\ if\ x(z,t) = 0 \tag{4}$$

are continuous functions along flow trajectories.

□

As a consequence, two vehicles entering a link at time t and t', respectively, $(t' > t)$ will leave the link at times τ and τ' and always $\tau' > \tau$, independently of their destination $q \in D$. First of results in previous lemma appears directly when approximating system (2) in the case of *constant propagation speed* ω, with an implicit regressive differences method of the type $x_j^{q,i+1} - x_j^{q,i} + \nu\omega \cdot (x_j^{q,i+1} - x_{j-1}^{q,i+1}) = 0$. Solutions of this implicit approximation can be represented by means of a grid time expanded network. Next lemma shows that amongst the set of paths in the grid time expanded network of a first order regressive implicit approximation it is always possible to find a subset of paths for which no crossing can occur between flows either of different or the same commodity, entering the link at different time subintervals. Therefore there will exist FIFO observance under the following two conditions: first, that an enough dense grid is used for the approximation of system (2) and second, that the CFL parameter $\nu\omega$ verifies $\nu\omega < 1/2$ (strengthened CFL condition).

LEMMA 2 (*FIFO observance for solutions of implicit first order approximation to multicommodity system* (2)) *Let system (2) in the case of constant speed propagation ω be approximated by an implicit method of the type:*

$$x_j^{q,i+1} - x_j^{q,i} + \nu\omega \cdot (x_j^{q,i+1} - x_{j-1}^{q,i+1}) = 0, \quad \nu = \frac{\delta_t}{\delta_z} \tag{5}$$

then two sufficient conditions in order to solution flows of (5) with different commodities do not intersect inside *a link are:*

1. *CFL parameter $\nu\omega < 1/2$.*
2. *An enough dense grid (δ_z, δ_t) is used for the approximation.*

□

Proof

When approximating (2) by means of a grid (δ_z, δ_t), link $a \in \mathcal{A}$ of the original network splits as shown in figure 1. Solution flows of (5) for different commodities $q, q' \in D$ can intersect each other *inside* link a either at an L-node or at an x-link. Intersecting flows a, α at L-nodes are linked by the following relationship $\alpha^q - a^q = x_0^q - v^q$ (R1) whereas flows a, α at x-links are linked by $\alpha^q - a^q = x_0^q - v_1^q - v_2^q$ (R2). Thus, for fixed x_0^q, v_1^q, v_2^q there exists a non unique decomposition of this magnitudes in a^q, α^q at both L-nodes and x-links. Then the no intersecting condition for flows of the same commodity can be stated as: 1) There exist α^q, a^q verifying (R1) such that $\alpha^q a^q = 0$ and the no intersecting condition at x-links or L-nodes for flows of different commodities can be stated as: 2) there exist α^q, a^q verifying (R2) such that $\alpha^q = 0, a^q \geq 0, \;\; \forall q \in D$ or, alternatively, 2') $\alpha^q \geq 0, a^q = 0, \;\; \forall q \in D$. Because of the *convergence* of method (5) to the exact solution of PDE system (2), as the grid is more dense $\alpha^q - a^q = x_0^q - v^q = x_0^q - \nu\omega x^q \rightarrow x_0^q(1 - \nu\omega) > 0$ if $\nu\hat{\omega} < 1$ and $\alpha^q - a^q = x_0^q - v_1^q - v_2^q = x_0^q - \nu\omega \cdot (x_1^q + x_2^q) \rightarrow x_0^q(1 - 2\nu\omega) > 0$, if $\nu\hat{\omega} < 1/2$ implying thus that there exists $\alpha^q \geq 0, a^q = 0$. □.

The final aspect that has to be considered is composition of exit flows at links in the multicommodity case. Previous lemma 1 states the fact that composition ratios are continuous functions on trajectories. This *homogeneity* is implicitely assumed for exit flows, i.e: $v^q(t) = \jmath^q(\hat{z}+, t) = r^q(\hat{z}-, t) \cdot \jmath(\hat{z}+, t)$. In the case of constant speed ω when system (2) is approximated by means of a finite difference method *homogeneity* must be *imposed* on exit flows. As the case of constant speed can not reproduce boundary conditions 4 unless by means of vertical queues at the end, additional relationships imposing homogeneity are needed for a realistic modelization:

$$\begin{aligned} &\kappa^{q,i+1} = \kappa^{q,i} + \delta_t(x_M^{q,i}\omega(x_M^{q,i}) - v^{q,i}) && (\textit{Vertical queues}) \\ &v^{q,i} = v^i \, \frac{\kappa^{q,i}}{\kappa^i}, \; \textit{if} \;\; \kappa^i > 0, \quad v^{q,i} = 0, \; \textit{if} \;\; \kappa^i = 0 && (\textit{Homogeneity condition}) \end{aligned} \tag{6}$$

However homogeneity condition at vertical queues, when expressed as in (6) leads to nonconvex formulations of the feasible set of flows of a dynamic traffic model, with the corresponding analytic and

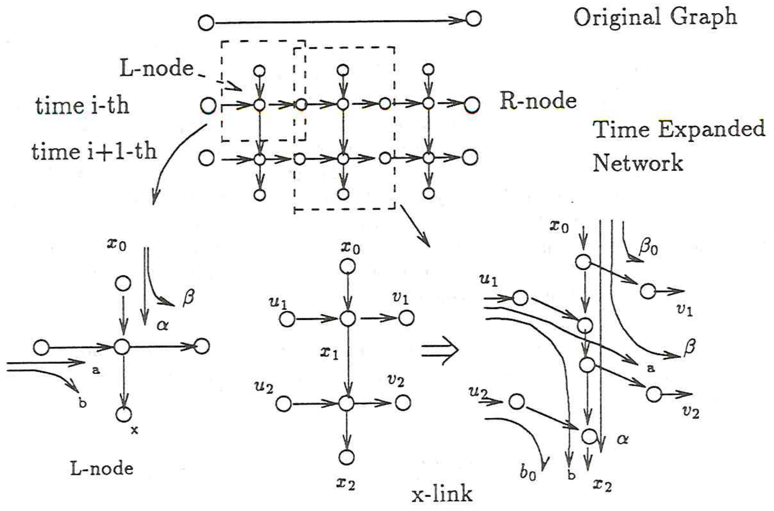


Figure 1: Time expanded network for link a of the original graph when an implicit first order regresssive approximation to system (2) is used and flow decomposition at x-links and L-nodes.

algorithmic difficulties in the resulting optimization problems. It is possible, however, to reformulate homogeneity condition (6) by means of the flow decomposition at L-nodes and x-links used in the proof of lemma 1 which ensures the existence of a subset of paths without crossing in the case of $x_0^q - v_1^q - v_2^q \geq 0$, v_1^q, $v_2^q \geq 0$. These *linear* inequalities must be imposed for each time subinterval and each commodity at the vertical queue of a link:

$$\begin{aligned} \kappa_{i-1}^q &\geq \delta_t (v_i^q + v_{i+1}^q), \quad i = 1, \dots N-1, \forall q \in D \\ \kappa_{N-1}^q &\geq \delta_t v_N^q, \qquad \forall q \in D \end{aligned} \tag{7}$$

and so, no crossing or selective output from the queue may exist.

5 First Order Approximations of Continuum Simple Model by an ODE System

In this section a brief summary of methods approximating PDE (1) in the case of constant speed-density relationships are examined accordingly to their stability. Stability requirements of the integration method used for the ODE system are important when approximating the System Optimal Dynamic Traffic Assignment Model by an Optimal Control Problem (see, for instance, [20]). We conclude with a simple example showing that for the case of decreasing speed-density functions adjusting to conditions stated in section 4, first order implicit methods used for the integration of an ODE system approximating PDE (1) may present *unstabilities*.

Approximations to PDE (1) by means of an ODE system can be made by M functions $x_k(t)$, $k = 1, \dots M$ that will be considered approximations to density $x(k\delta_z, t)$, i.e. the exact solution of PDE (1) at points $z_k = k\delta_z$ and M functions $\jmath_k(X)$, $k = 1, \dots M$, approximations to the flow $\jmath(k\delta_z, t)$. We will assume $\jmath_k(X) = \jmath_k(x_k)$. Let $X^\top = (x_1(t), x_2(t), \dots, x_M(t))$ and $J^\top = (\jmath_1(X), \jmath_2(X), \dots, \jmath_M(X))$. We will assume $\jmath_k(X) = \jmath(x_k) = x_k \omega(x_k)$. Then PDE (1) can be approximated by means of:

$$\dot{X} = -\frac{1}{\delta_z} CJ(X) + b(t) \tag{8}$$

Where C is a constant matrix that depends of the type of approximation made to the term $\frac{\partial \jmath}{\partial z}$ (see for instance, [22]) and $b(t)$ is related with the boundary conditions at $z = 0$ and $z = \hat{z}$. Initial conditions for ODE system (8) are given by values of the density at $t = 0$, $x_k(0) = h(k\delta_t)$. Then, it is known that if $\jmath(x_k) = \omega \cdot x_k$ (constant speed $\omega > 0$) the following approximations to the term $\frac{\partial \jmath}{\partial z}$ lead to ODE approximating systems and that these systems can be solved by an implicit first order method presenting *numerical stability*:

1. First order regressive differences. $\frac{\partial \jmath}{\partial z} = \frac{\jmath_k - \jmath_{k-1}}{\delta_z} + O(\delta_z)$
2. Double regressive second order differences. $\frac{\partial \jmath}{\partial z} = \frac{3\jmath_k - 4\jmath_{k-1} + \jmath_{k-2}}{2\delta_z} + O(\delta_z^2)$
3. Mixed central-progressive first order differences. $\frac{\partial \jmath}{\partial z} = \frac{2\lambda \jmath_k - (1+\lambda)\jmath_{k-1} + (1-\lambda)\jmath_{k+1}}{2\delta_z} + O(\delta_z)$. being $\lambda > 1$ the degree of regressivness.
4. Mixed regressive-progressive first order differences. $\frac{\partial \jmath}{\partial z} = \frac{(2\lambda-1)\jmath_k - \lambda \jmath_{k-1} + (1-\lambda)\jmath_{k+1}}{\delta_z} + O(\delta_z)$. being $\lambda > 1$ the degree of regressivness.

However, approximation by central differences $\frac{\partial \jmath}{\partial z} = \frac{\jmath_{k+1} - \jmath_{k-1}}{2\delta_z} + O(\delta_z^2)$ may lead to an ODE system of the type (8) with solutions highly unstable when compared with the exact solutions of PDE (1) as shown in [22].

Stability is lost when speed-density relationships are non constant but decreasing as required in section 4. If ODE system (8) is solved by an implicit method:

$$X^{i+1} - X^i = -\nu CJ(X^{i+1}) + b^{i+1}, \quad (\nu = \frac{\delta_t}{\delta_z}) \tag{9}$$

then, main part of the errors ϵ will propagate following the relationship $\epsilon^i = (I + \nu C\Phi^i)^{-1}(I + \nu C\Phi^{i-1})^{-1} \ldots (I + \nu C\Phi^1)^{-1}\epsilon^0$, being $(\Phi^i)^\top = (\varphi(x_1^i), \ldots, \varphi(x_M^i))$ $(\varphi(x) = \omega(x) + x\omega'(x)$, the derivative of the flow-density relationship). Unfortunately in many cases $\|(I + \nu C\Phi^i)^{-1}\|_2 > 1$ and then $\|\epsilon^i\|_2$ grows as i increases. Consider, for instance, the case $M = 1$, an approximation of the term $\frac{\partial \jmath}{\partial z}$ by any of the previous schemes and an *overcritical* solution sequence $x^i > \bar{x}$, then $\Pi_{j=1}^i (1 + \nu c\varphi^j)^{-1}$ grows as i increases because $c\varphi^j \leq 0$ (being c the (positive) constant for $\jmath_k^i$ in the approximation of the term $\frac{\partial \jmath}{\partial z}$ at $z = k\delta_z$, $t = i\delta_t$).

6 A System Optimal Model with Distributed Parameters.

State equations of Merchant and Nemhauser's Model can be considered an approximation to the Simple Continuum Model. This model reproduces basic traffic behaviour characteristics: finite propagation speed of flows, dispersion, backward propagation or spillback and no intersection between trajectories even when flows are composed by a set of commodities. For an initial distribution of densities, input and exit flows on a link determine univoquely flow propagation and can be considered as controls. Therefore a generic System Optimal Dynamic Traffic Assignment model with separability of the costs consists of the minimization of a given functional of the nonnegative solutions of systems (2). Solutions of these systems are linked each other by means of input and exit flows on links verifying these input and exit flows a set of balance equations at nodes of the original graph. In this section after an exposition of the model, it is shown how it can be approximated by an optimal control problem (OCP) in the case of constant speed ω_a and vertical queues at links.

A System Optimal Dynamic model can be formulated as:

DOCP 1 (*Multiple destination optimal control with distributed parameters. Extension of* System Optimal *Merchant and Nemhauser's model.*)

$$\begin{array}{lll}
Min_{x_a,\jmath_a} & \sum_{a\in A}\int_0^T\int_0^{\hat{z}_a} F_a(\, x_a(z_a,t), \jmath_a(z_a,t), z_a, t\,)\, dz_a\, dt & + \int_0^T S_q(s^q(t))\, dt \\
s.a & \dfrac{\partial x_a^q}{\partial t} + \dfrac{\partial \jmath_a^q}{\partial z_a} = 0 & (\,\forall a \in \mathcal{A}\,) \\
 & B_+\jmath^q(0,t) - B_-\jmath^q(\hat{z},t) + e_q s^q(t) = p^q(t) & \\
 & \jmath_a^q(z_a,t) = x_a^q(z_a,t)\cdot\omega_a(\, x_a(z_a,t)\,) & \forall q \in D \\
 & \jmath^q(z_a,t) \geq 0 & (\, z_a \in [0,\hat{z}_a], \;\; \forall a \in \mathcal{A}\,) \\
 & (\, x_a(z_a,t) = \sum_{q\in D} x_a^q(z_a,t)\,) & \\
 & (\, \textit{Known initial state } x_a^q(z_a,0)\,) &
\end{array}$$

In the case of constant propagation speed ω and vertical queues at the end of links, it is possible the approximation of model **DOCP 1** by means of an optimal control problem as numerical stability of the resulting state equations is ensured. Approximation of systems (2) can be made by means of discretizing each "z-continuous" link a in M_a sublinks and approximating systems (2) by means of systems of ODE's as described in previous section. For instance, in the case of a first order regressive approximation of the term $\frac{\partial \jmath}{\partial z}$ the approximating ODE system would be:

$$\begin{pmatrix} \dot{x}_1^q \\ \dot{x}_2^q \\ \vdots \\ \dot{x}_{M_a}^q \\ \dot{\kappa}_a^q \end{pmatrix} = -\frac{\omega_a}{\delta_{z_a}} \begin{pmatrix} 1 & & & & 0 \\ -1 & 1 & & & 0 \\ & -1 & \ddots & & \vdots \\ & & \ddots & 1 & 0 \\ & & & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1^q \\ x_2^q \\ \vdots \\ x_{M_a}^q \\ \kappa_a^q \end{pmatrix} + \frac{1}{\delta_{z_a}} \begin{pmatrix} u_a^q(t) \\ 0 \\ \vdots \\ 0 \\ -v_a^q(t) \end{pmatrix} \tag{10}$$

or in more compact form:

$$\dot{y}_a^q = -\varrho_a\, C y_a^q + b_a^q(t), \quad (\varrho_a = \frac{\omega_a}{\delta_{z_a}}\,) \tag{11}$$

verifying input and exit flows at links the following set of balance equations:

$$B_+u^q(t) - B_-v^q(t) + e_q s^q(t) = p^q(t) \tag{12}$$

Having into account a limited capacity at vertical queues and imposing homogeneity on exit flows from the vertical queue, problem **DOCP 1** can be approximated by the following OCP (optimal control problem):

OCP 1

$$\begin{array}{ll}
Min_{\,y,u,v} & \sum_{a\in\mathcal{A}} \left\{ \delta_{z_a} \sum_{j=1}^{M_a} \int_0^T F_a(x_{j,a}, \omega_a x_{j,a})\, dt \right\} \\
s.t. & \dot{y}_a^q = -\varrho_a\, C y_a^q + b_a^q(t) \\
 & B_+u^q - B_-v^q + e_q s^q = p^q(t) \\
 & \kappa_a v_a^q = \kappa_a^q v_a \kappa_a \\
 & 0 \leq \kappa_a \triangleq \sum_{q\in D} \kappa_a^q \leq \hat{\kappa}_a \\
 & v_a^q \geq 0, \;\; u_a^q \geq 0, \;\; s^q \geq 0 \\
 & (\, \textit{Known initial state } x_{a,j}^q(0)\,)
\end{array}$$

If this OCP is discretized in t with step δ_t, observing the strengthened CFL condition at each link, i.e. $\omega_a \frac{\delta_t}{\delta_{z_a}} < 1/2, \forall a \in \mathcal{A}$, as required by lemma 1, homogeneity conditions at vertical queues are expressed as in (7), and first order regressive approximation to $\frac{\partial \jmath}{\partial z}$ are used, then the following optimization problem arises:

ΔOCP 1

$$
\begin{array}{llll}
Min_{\,y,u,v} & \delta_t \sum_{a\in\mathcal{A}} \delta_{z_a} \sum_{j=1}^{M_a} \sum_{i=1}^{N} F_a(x^i_{j,a}, \omega_a x^i_{j,a}) & (L.M.) & \\
s.t. & y^{q,i}_a - y^{q,i-1}_a + \nu_a \omega_a y^{q,i}_a - b^{q,i}_a = 0, & \lambda^{q,i}_a & \nu_a = \dfrac{\delta_t}{\delta_{z\,a}} \\
 & B_+ u^{q,i} - B_- v^{q,i} + e_q s^{q,i} = p^{q,i} & & i = 1, \dots N \\
 & \kappa^{q,i-1} \geq \delta_t (v^{q,i} + v^{q,i+1}) & \varsigma^q_i, & i = 1, \dots N-1 \\
 & \kappa^{q,N} \geq \delta_t v^{q,N} & \varsigma^q_N & \\
 & u^{q,i} \geq 0, \; v^{q,i} \geq 0, \; s^{q,i} \geq 0 & & \\
 & 0 \leq \sum_{q\in D} \kappa^q_a \leq \hat{\kappa}_a & \mu^i_a, \; \bar{\mu}^i_a & \\
 & (\; Known \;\; initial \;\; state \;\; x^q_{a,j}(0)\;) & & \\
 & (\; \delta_t < Min_{a\in\mathcal{A}} \left\{ \dfrac{\delta_{z_a}}{2\omega_a} \right\}\;) & &
\end{array}
\tag{13}
$$

Difference state equations of this optimization problem are now a *convergent* approximations to solutions of systems (2). It must be emphasized that all constraints are *linear*. We shall consider now the case of *strict* convex costs functions $F_a(x_a, \jmath_a)$. It is possible then the partial dualization of homogeneity constraints and limited capacity at vertical queue constraints $0 \leq \sum_{q\in D} \kappa^q_a \leq \hat{\kappa}_a$. resulting the dualized objective:

$$
\begin{aligned}
\Psi(x, \omega\cdot x, \varsigma, \mu, \bar{\mu}) = \; & \delta_t \sum_{a\in\mathcal{A}} \delta_{z_a} \sum_{j=1}^{M_a} \sum_{i=1}^{N} F_a(x^i_{j,a}, \omega_a x^i_{j,a}) - \sum_{i=1}^{N-1} \varsigma^{q\,\top}_i (\kappa^{q,i-1} - \delta_t(v^{q,i} + v^{q,i+1})) + \\
& + \varsigma^{q\,\top}_N (\kappa^{q,i-1} - \delta_t v^{q,i}) - \sum_{i=1}^{N} \mu^{i\,\top}_a \kappa^i_a + \sum_{i=1}^{N} \bar{\mu}^{i\,\top}_a (\hat{\kappa}^i_a - \kappa^i_a)
\end{aligned}
\tag{14}
$$

The case of a single destination. In the case of a single destination in the network it is possible the application of a decomposition method developed in [9] for nonlinear programming problems approximating optimal control problems, which is equivalent to an implicit integration method for the extremals of such problems. Given a nonlinear optimization problem approximating an optimal control problem discretized in N time subintervals ($\delta = T/N$) of the type:

ΔOCP 2

$$
\begin{array}{lll}
Min_{\,x_i,\,u_i} & \delta \sum_{i=1}^{N} f_0(x_i, u_i) & (L.M.) \\
s.t. & \delta A u_i + x_{i-1} - x_i = 0 & \lambda_i \\
 & C u_i = D(t_i) & \alpha_i \\
 & u_i \geq 0 & \varsigma_i \\
 & (x_0 \; known), \;\; (i = 1, \dots, N) &
\end{array}
$$

The decomposition method results in the solution of successive nonlinear optimization subproblems of the type Δ **OCP2 1** shown below, each of them for a time subinterval $[(i-1)\delta, i\delta]$. In [9] it is proved how first order conditions of these subproblems approximate the extremals of an optimal control problem provided that subproblems present *sensitivity* of their solutions for a certain parameters λ' which are proved to be the an approximation to the adjoint variables of the optimal control problem at time $(i+1)\delta$.

ΔOCP2 1

$$\begin{array}{llll} Min_{\,x,u} & \delta f_0(x,u) - \lambda'^{\mathsf{T}} x & & (L.M.) \\ s.t. & \delta A u + x_{k-1} - x \;=\; 0 & & \lambda(\lambda') \\ & Cu \;=\; D(t_i) & & \alpha(\lambda') \\ & u \geq 0 & & \varsigma(\lambda') \end{array}$$

We write below the structure of subproblems Δ **OCP2 1** when the decomposition method is applied to problem Δ **OCP 1**:

ΔOCP2 2

$$\begin{array}{lll} Min_{\,y,u,v} & \psi_i(x,\omega x,\varsigma,\mu,\bar{\mu}) \;-\; \lambda'^{\mathsf{T}} y^i & \\ & & \\ s.t. & y_a^i - y_a^{i-1} + \nu_a \omega_a y_a^i - b_a^i \;=\; 0 & \\ & B_+ u^i - B_- v^i + e_q s^i \;=\; p^i & \\ & u^i \geq 0,\; v^i \geq 0,\; s^i \geq 0 \qquad\qquad i \;=\; 1,\ldots N & \end{array} \tag{15}$$

with the objective function ψ_k containing only terms of the right of (14) corresponding to a given time superscript k.

In [9] and [7] it is shown the network structure of the linear constraints of Δ **OCP2 2**. Additionally, for the case of networks with a single destination, if cost functions $F_a(x,\jmath)$, depending on the total flow $\jmath$ and density x, are *strictly* convex on $\jmath$ and x it is possible to guarantee sensitivity of solutions of problem Δ **OCP2 2** respective parameter λ'. Unfortunately, the use of this decomposition method requires that subproblems Δ **OCP2 2** be solved with accuracy enough because, as shown in [9], errors propagate as the extremal calculation progresses. Submanifold optimization algorithm (see, for instance [24]) seems a suitable candidate in the case of quadratic positive definite costs to overcome accuracy requirements of this decomposition method.

7 Remarks

This paper presents a generic System Optimal Dynamic Traffic Assignment model for multidestination networks based on an extension of the simple continuum model. This model can be considered as a reference because of basic characteristics of its flow dynamics, no overtaking between flows of different destinations, finite speed propagation, dispersion and effects of congestion such as spillback. For the case of constant propagation speed and vertical queues at the end of links numerically stable approximations to the system optimal model by means of optimal control are presented. For the flow dynamics based on regressive difference approximations of the term $\frac{\partial \jmath}{\partial z}$ it is possible the formulation of a nonlinear programming problem with *linear* constraints, thus avoiding nonconvexities of constraints for FIFO observance in previous formulations.

References

[1] Addison J.D., Heydecker B.G. (1992) *"Traffic Models for Dynamic Assignment."* Paper presented to the Second International Capri Seminar on Urban Traffic Networks, Capri, July 1992.

[2] Anderson D.A., Tannehill J.C., Pletcher R.H (1984) *"Computational Fluid Mechanics and Heat Transfer."* Hemisphere Publishing Corporation 1984

[3] Bryson A.E. *"Applied Optimal Control "* John Wiley & Sons New York (1975)

[4] Carey M. (1986) *"A Constraint Qualification for a Dynamic Traffic Assignment Model"* Transportation Science, 20, pp 55-58

[5] Carey M. (1987) *"Optimal Time-Varying Flows on Congested Networks."* Operations Research Vol.35, No 1, January-February 1987.

[6] Carey M. (1991) *"Nonconvexity of the Dynamic Assignment Problem"* Transpn Res.B Vol 26B, No 2, pp. 127-133

[7] Codina E., Barceló J. (1992) *"Extremals Calculation for the Dynamic Traffic Assignment Problem "* Presented at 39th North American Meeting of the RSAI, Chicago USA 1992

[8] Codina E., Barceló J. (1992) *'Dynamic Traffic Assignment: An Annotated Review of Modelling and Algorithmic Approaches "* Universitat Politècnica de Catalunya. Departament d´Estadística i Investigació Operativa Document de Recerca DR 94-01

[9] Codina E. (1993) *"Un Algoritmo para Problemas de Control Optimo y su Aplicación a la Asignación Dinámica de Tráfico"* Tesis Doctoral. Universitat Politècnica de Catalunya. Departament d´Estadística i Investigació Operativa

[10] T.L.Friesz, F.J.Luque, R.L.Tobin, B.W.Wie (1989) *"Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem"* Op. Research Vol 37-6 pags 58-69 USA

[11] Ho J.K., Manne A.S. (1974) *"Nested Decomposition for Dynamic Models"* Math. Prog. 6, 121-140 (1974)

[12] Ho J.K. (1980) *"A succesive Linear Optimization Approach to the Dynamic Traffic Assignment Problem"* Transportation Science 14 295-305.

[13] Ho J.K.(1990) *"Solving the Dynamic Traffic Assignment Problem on a Hypercube Multicomputer "* Transportation Research Vol 24B 6 1990.

[14] Luque F.J., Friesz T.L. *"Dynamic Traffic Assignment Considered as a Continuous Time Optimal Control Problem "* Presented at the TIMS/ORSA Joint National Meeting, Washington, D.C., May 5-7, 1980

[15] Merchant D.K., Nemhauser G.L. (1978) *"A Model and an Algorithm for the Dynamic Traffic Assignment Problems."* Transportation Science Vol.12 No 3 August 1978.

[16] Merchant D.K., Nemhauser G.L. (1978) *"Optimal Conditions for Dynamic Traffic Assignment Model."* Transportation Science Vol.12 No 3 August 1978.

[17] Mitchel A.R., Griffiths D.F. (1980) *"The Finite Difference Method in Partial Differential Equations"* John Wiley and Sons, New York (1990)

[18] Papageorgiou M. (1989) *"Dynamic Modeling, Assignment, and Route Guidance in Traffic Networks "* Transpn Res. Vol 24B, No 6, pp. 471-495 (1990) (From US Italy Joint Seminar Preprints at Capri June 1989)

[19] Ran B., David E.B., LeBlanc L.J. (1990) *"A New Class of Instantaneous Dynamic User-Optimal Traffic Assignment Models"* Operations Research Vol. 41, No. 1993

[20] Sage A.P., White C.C. III *"Optimum Systems Control "* Prentice Hall, Englewood Cliffs, New Jersey (1977)

[21] Shames I.H. (1962) *"Fluid Mechanics"* Mac Graw Hill (1962)

[22] Smith G.D. (1985) *"Numerical Solutions of Partial Differential Equations: Finite Difference Methods"* 3rd edition. Oxford Applied Mathematics and Computing Science Series

[23] B.W.Wie, T.L.Friesz, R.L.Tobin (1990) *"Dynamic User Optimal Traffic Assignment on Congested Multidestination Networks "* Transportation Research 24B 6 December 1990

[24] Zangwill W.I. (1969) *"Nonlinear Programming: A Unified Approach."* Prentice-Hall, Englewood Cliffs, NJ, 1969.

DYNAMIC ASSIGNMENT MODELLING ON CONGESTED NETWORKS

Giuseppe Bellei
Dipartimento di Idraulica, Trasporti e Strade
Università degli Studi di Roma "La Sapienza
"Via Eudossiana 18 - 00184 Roma, Italia

Maurizio Bielli
Istituto di Analisi dei Sistemi ed Informatica Consiglio Nazionale delle Ricerche
Viale Manzoni 30 - 00185 Roma, Italia

INTRODUCTION

The model and assignment technique presented carry out previous work on mixed (discrete/continuous) heuristic approach to within-day dynamic route choice, where a discrete representation is adopted for departure time and a continuous one for arrival time at nodes and destinations.

Several continous dynamic traffic assignment problem formulations have been proposed by, among others, Friesz et al. (1989), Ran and Shimazaki (1989), Wie et al. (1990), Boyce et al (1992). They derived continous equilibrium models as optimal control problems based on the principle of selfish path time minimization.In these works path times were evaluated as the sum of current link times at the moment path choice is performed or revised, so defining an instantaneous user optimum, or reactive equilibrium in the terminology of Papageorgiu (1990), who first acknowledged the distinction between the two forms of equilibrium conditions arising in a dynamic framework, naming predictive equilibrium the one based on link times to be encountered when travelling.

Vythoulkas (1990) presented, generalizing previous results obtained by Ben-Akiva et al. (1984) on a parallel links network, another continous route and departure time choice model, not making any explicit reference to optimal control theory and defining, in analogy with the static case, stochastic predictive equilibrium conditions as a steady state for an implicit Markovian model of day to day demand adjustment; (numerical results are however obtained from a partially discretized version of the problem).

Recently Ran et al. (1992a,1992b) defined and studied, still in an optimal control theory framework, also the properties of continous models for an "ideal dynamic user optimum", that is for predictive equilibrium, extending these models to the choice of time when to travel and to the case of stochastic route travel times. Nevertheless, they didn't present for this last models any solution algorithm, while it should be noted that also for reactive equilibrium the solution algorithms and numerical result available are obtained by developing a fully discretized version of the problem and testing it on small "toy" networks.

On the other side, some kind of discretization was adopted in the earliest attempts to define a dynamic traffic assignment model, from the pioneering simulation models by Yagar (1976), to the first mathematical programming models by Merchant and Nemhauser (1978a,1978b), dealing with single destination, system optimal route choice. This kind of models was studied and perfectioned in more recent years by several authors, like Carey (1986,1987) and Ho (1980,1990); they were also extended to time dependent departure patterns and tested on a practical application to a corridor network by Mahmassani and Chang (1986,1988), always keeping a discrete time representation. Carey (1992) also pointed out the relevance of no overtaking conditions in dynamic traffic assignment.

A significantly different approach was taken by Cascetta and Cantarella (1990), who developed the Markovian model implicitly assumed by Vythoulkas and obtained a comprehensive non-equilibrium description of traffic phenomena.

In another group of models, time varying traffic flows are determined by directly applying an assignment technique defined in analogy with the static case, like in Hamerslag (1988), or by heuristically solving a dynamic assignment problem in its turn defined by analogy, like in Bellei and Bielli (1990,1992) and Janson (1991). It is however only from heuristics and simulation approaches that came, up to now, numerical results on full size networks.

The model and dynamic assignment technique which have been developed are based on a model for link performance. The model takes into account queueing and allow checking of spill-back to verify the magnitude of the phenomenon, while however approximately accounting for involved delays (in current version) and indirectly representing backward spreading of queues by iteratively adjusting link outflow capacity (as a programmed development). At first, this link model is developed independently by the network framework where within day dynamic assignment of travel demand is carried on to represent daily route choices. The inflow is assumed as given and piecewise constant. Outflow has an upper limit, like the number of vehicles queueing on the link, wich is thus a bottleneck with known outflow and storage capacity.

Notwithstanding this limitations, which are however, at least in part, easy to overcome (developing extensions to the proposed technique and/or expanding network representation) this link performance model is able to give a picture of traffic flowing on congested link, supplying a quantitative description of what happens to any entering vehicle, be it smoothly travelling from initial to final node, or held for some time in a queue. The presentation begins with the description of this model, while the network model in which it is framed, the assignment technique developed, guidelines for further development and some preliminary numerical results are dealt with in this order.

LINK MODEL

With regard to the link performance model it is assumed that, on any element of a road network represented by a link, vehicles may be only in one of the following two conditions: travelling at a constant speed V, eventually a suitable fraction of free flow speed, or queueing along the link. If inflow entering the link is lower than its outflow capacity OC (eventually a suitable fraction of actual link capacity as determined by physical capacity and green splits), all vehicles travel at speed V, no queueing take place and vehicles exit the link at the same rate they enter it, otherwise vehicles exit the link at a constant outflow C and the queue occupies a section of the link, going backward from final node for a variable in time length (queue occupancy) OQ, determined by average spacing between queueing vehicles D, number of lanes M and variable-in-time entering flows F^i, assumed as constant within each time interval i from time T^{i-1} to time T^i. Queue occupancy OQ has a maximum, corresponding to link length L, proportional to overall storage capacity SC (maximum number of vehicles queueing on a link), given by L·M/D.

At first, the limiting assumption that queue occupancy is non zero and shorter than link length is made and a linear arrival time function is derived for each interval. In this way spill-back queues don't need to be considered, link performances are independent from each other and both queue occupancy and arrival times at the end of the link can be determined as functions of the vector of link inflows **F** and of link entring time t. These functions OQ(**F**, t), TA(**F**, t) result to be, with the assumptions made, linear within each interval i and piecewise linear with respect to t on the whole time horizon considered. Moreover, they can be recursively determined, with the only requirement that an initial state of the link (in practice an uncongested state) is known.

Also the event of queue's vanishing during some interval i is easily checked and a zero occupancy may be attributed to the queue whenever negative values would be obtained by recursions and linear relationships. Resulting queue occupancy and arrival times keep being piecewise linear functions of entering time, but a breakpoint is introduced within the interval.

In case the queue should reach, on some link, its maximum length L, it should spill back into links having initial node of that link as their final node, so that interaction between link performances would take place; an approximate representation of link congestion phenomena may avoid taking into account such an interaction by considering vehicles which can't be accommodated along the link (on the "horizontal" queue) as waiting into a fictitious tank where a FIFO rule is respected (in other words, a "vertical" queue) located at initial node. This approximate representation doesn't require any substantial change in recursive definition of arrival times with respect to the case of queue occupancy not exceeding link length, while fictitious queue occupancies, or total number of queueing vehicles, have to be updated to identify times when storage capacity exceeding and recovering takes place.

The arrival time function defined by link performance model has the interesting property of having non negative derivatives with respect to each of its arguments, so that a no overtaking constraint is implicitly satisfied and paradoxical results like the decrease of some travel times for increasing average inflows are avoided. Other approaches focusing on queues representation to deal with link performance in a dynamic assignment framework can be found in Drissi-Kaïtouni and Hameda-Benchekroun (1992) and in Bernstein et al. (1993).

NETWORK MODEL

Link performance model plays in a dynamic framework the role that performance functions play in static equilibrium models, representing the behaviour of an element of transportation system when it has to carry a given traffic load. Hence a network model has to be built to determine the traffic load for each element, which has to be consistent with link performance model and able to define the time, space and flow relationships among elements, in such a way that transportation offer over the whole network, as well as relationships between transportation offer and demand, are represented.

Space relationships among elements of transportation system are represented as usual as a directed graph G(N, L) where paths are defined as an ordered sequence of nodes $n \in N$, from an origin $r \in R$ to a destination $s \in S$ (R,S subsets of N), such that no node appears more than once and a link $mn \in L$ exists for any two consecutive nodes in the sequence.

Time is represented in different ways for different purposes, both as a non negative real variable and as intervals, whose definitions are introduced to allow two important simplifying assumptions:

a) departures taking place during any interval j $(TD^{j-1}, TD^j]$ are grouped, to represent vehicles travelling on the network as platoons departing together at some fixed time (i. e. time TD^j for interval j);

b) average flow determined by platoons arriving at a node m during any interval i $(TN^{i-1}, TN^i]$ and travelling on link mn is dealt with as a constant inflow, entering the link during the interval.

Even if different intervals may be defined for each origin and node, only intervals with respect to departure from origins and to arrival at nodes are distinguished at first, identified by times TD^j and TN^i respectively. The time of the day dimension of a trip on path p, that is the "journey" corresponding to that trip (Addison and Heydecker, 1993), is described by non negative real arrival times $TA^j_{mp}(\Phi)$ at node m, when departing at TD^j and travelling along p, derived with recursive compounding from arrival time functions and thus depending from the vector Φ of inflows for each interval and link.

These inflows are the same introduced when dealing with link perfomance model as a vector **F**, conceptually defined by assumption b) above. Path flows FD^j_p for any departure time TD^j are also defined to represent route choice by travellers departing from the origin of p during interval j, while inflows are the only link flows which are explicitly defined, since space and time continuity of flows is guaranteed if path flows and link inflows are consistent, in the sense that they satisfy a subproblem of dynamic assignment commonly known as dynamic network loading.

Transportion demand is simply defined as a three dimensional OD matrix given in terms of number of vehicles ND^j_{rs}, departing from origin r to destination s at time TD^j, having grouped, consistently with assumption a), flows GD^j_{rs} departing during interval j in platoons departing at time TD^j.

The relationship between path flows (defined with respect to departure time) expressing route choice and consistency with travel demand, and link inflows (defined with respect to arrival time at nodes) loaded to the network and determining its performance, is a distinctive feature of time of the day dynamic assignment.

The definition of this relationship requires formulation of dynamic network loading problem, but it is in any case worth noting that a difference between flows defined with respect to different times arises because the number of veichles departing from an origin r to travel along a path p during a time interval do not pass through nodes within the same time interval, so that direct relationships between mobility demand and network flows, usually adopted in the static case, are no longer exhaustive in representing mobility phenomena.

Moreover, even if departure time flow were assumed as constant along interval j, their contribution to link inflows would be influenced, at any link, by changes in flows sharing that link during the time needed to go from initial to final node. It is easily acknowledged that, due to additive and multiplicative effects when passing from link to link, the "history" of departure time flows on any non trivial network is too complex to be explicitly represented, so that some kind of approximation, avoiding such a representation, but allowing to distinguish betwen flows defined with respect to departure time and node arrival time, is needed. An approach explicitly dealing with history of vehicle platoons travelling the network has been developed, in a fully discretized framework, by Smith (1992).

Such an approximation is supplied in this framework by assumptions a) and b) above. To accept approximation a), the difference in traffic conditions encountered by the vehicles loosing their identity to become part of a compact platoon must be negligible, as it happens when departure time interval j and/or platoon size ND^j_p are small enough. In principle this condition is easily satisfied, because, if platoons are defined as ND^j_p, their size is negligible for all practical purposes. To accept approximation b) interval length has to be assumed small enough to make averaging effect negligible.

Formally the average inflow into link mn during interval i can be expressed dividing the sum of the number of vehicles departing at time TD^j, choosing to travel along any path p including link mn and arriving at node m during interval i, by interval length.

Thus arrival times at nodes are needed to define link inflow vector Φ; in other words, denoting by $\mathbf{T}$ the vector of arrival times at each path's node, it is $\Phi = \Delta(\mathbf{T})$ or, if dependence from the vector of path flows for any origin destination pair and departure time Π is made explicit, $\Phi = \Delta(\Pi, \mathrm{T})$.

Since, as it has been noted earlier, arrival times depend in their turn from link inflows, it is evident how a circular dependence similar to the one existing between flows and times at equilibrium in static assignment takes place. Denoting by $\mathbf{T} = \mathrm{T}(\Phi)$ the relationship between inflows and node arrival times supplied by recursive application of link performance model, the fixed point formulation, specific to this model, of what is commonly known as the dynamic network loading problem, is to determine link inflows such that $\Phi = \Delta[\Pi, \mathrm{T}(\Phi)]$, where path flow vector is assumed as given.

As it has been noted with reference to link performance model, arrival time functions, being monotone non decreasing with respect to link entering time, implicitly satisfy a no overtaking constraint at link level. This constraint, because path arrival times at nodes are compound functions of link arrival time functions is also satisfied for each path. Compounding preserves also other interesting properties of arrival time function, but definition of $\mathrm{T}(\Phi)$ as a path-node-departure interval dimension vectorial function would create some difficulty, mainly if explicit path enumeration is needed, even if attempts has been made, like the one of Di Gangi (1992), to deal with path formulation of dynamic network loading.

To avoid path enumeration, compact platoons must be defined at a more aggregate level (care should be taken, in this case, to define departure intervals in such a way that relevant platoons' size results to be negligible). If aggregation is perfomed with respect to path and destination the flow determined by vehicles departing from origin r at time TD^j and choosing to travel along any path including link mn can be defined, but inflows can't be derived, in general, from these link-origin choice flows, since arrival times at nodes may differ among paths. If minimum arrival times at nodes **TM** for a certain vector **T** of arrival times, uniquely defined for each origin, are utilised, instead of actual arrival times on paths, an aggregate and approximate network loading would be determined.

It should be noted that, while average inflows $\Phi = \Delta(T)$ would be actually loaded to links mn in correspondence to intervals i if node arrival times were given by **T**, flows $\Phi M = \Delta(TM)$ even if indirectly determined by node arrival times, would differ from actual flows unless only minimum time paths are chosen. As it will be stated more formally in the following, however, if an equilibrium state for users choices is attained, travel times from the same origin and departure time to network nodes must be the same on any utilized path, so that the difference between the two above defined flows would vanish. Hence, if an equlibrium is reached through aggregate (link-origin level) representation of route choice it will be such also for a disaggregate (path level) one.

An iterative adjustment procedure aimed at verifying the fixed point condition stated above is thus applied to flows ΦM, instead of Φ, and different versions of this procedure are developed and tested. Assuming as negligible the approximation introduced in such a way, the arrival times **T** may be considered as argument of a shortest path in time assignment mapping supplying a vector of path flows $\Pi = A(T)$.

If dynamic network loading problem is assumed as solved, the resulting flow vector Φ can be considered as a function of path flows Π appearing as argument of function $\Delta(\cdot)$. Denoting by $\Lambda(\Pi)$ such a function, dynamic assignment problem in terms of path flows may be expressed as the generalized fixed point problem of finding Π such that $\Pi \subseteq A\{T[\Lambda(\Pi)]\}$ and equilibrium conditions may be defined as an almost straightforward extension to a time of the day dynamic framework of Wardrop's first principle requiring equal minimum travel times on utilised paths.

The dynamic assignment technique developed has a four level structure:

1) Arrival time function definition from given inflows;
2) Approximate dynamic network loading of given path choice flows, developed in different versions and implying arrival time functions adjustment;
3) Determination of path choice flows on the basis of current arrival time functions and averaging with previously determined path choice flows;
4) Adjustment of outflow capacities to eliminate vertical queues and take into account spill-back.

The first level corresponds to performance functions calculation in static assignment, the second is a testing of different degrees of aggregation and approximation in iteratively solving dynamic network loading, the third corresponds to the application of a successive averaging method, converging to exact solution of static assignment problem, as an heuristic for dynamic assignment. The last level is in course of development, hence some of the assumptions made have to be relaxed, with respect to current exposition, (i. e. outflow capacity have to be taken as variable among arrival intervals, which in their turn can differ among links) and an expanded network representation with turning links is needed to allow capacity adjustment to be determined for upstream links when storage capacity exceeding takes place down stream. Some preliminary result is presented with regard to the ability of assignment technique to find equilibrium flow and time patterns on small size networks.

REFERENCES

Addison J.D. and Heydecker B.G. A Mathematical Model For Dynamic Traffic Assignment *Transportation and Traffic Theory C. F. Daganzo ed. pp 171-183 Elsevier* **1993**

Bellei G., M Bielli Dynamic Assignment for Trip Planning Systems Assessment *Atti del Convegno Nazionale ANIPLA L'Automazione nei Sistemi di Trasporto* **1990**

Bellei G., M Bielli Sensitivity Analysis of a Dynamic Equilibrium Model for Route and Arrival Time Choice *presented at the 2nd Capri Seminar on Urban Traffic Networks* **1992**

Ben-Akiva M., M. Cyna, A. De Palma Dynamic Model of Peak-period Traffic Congestion *Transp. Res. 18B pp.339-355* **1984**

Bernstein D., T. L. Friesz, R. L. Tobin, B. W. Wie A Variational Control Formulation of the Simultaneous Route and Departure-Time Choice Equilibrium Problem *Transportation and Traffic Theory C. F. Daganzo ed. pp 107-126 Elsevier* **1993**

Boyce D.E., B. Ran L.J. LeBlanc Solving Dynamic User Optimal Traffic Assignment Model *UrbanTransportation Center Advance Working Papers Series N.11 (submitted to Transp. Sci.)* **1992**

Carey M. A Constraint Qualification for a Dynamic Traffic Assignment Model *Transp. Sci. 20 pp.55-58* **1986**

Carey M. Optimal Time-varying Flows on Congested Networks *Oper. Res. 35 pp.58-69* **1987**

Carey M. Nonconvexity of the Dynamic Traffic Assignment Problem *Transp. Res. 26B pp. 127-133* **1992**

Cascetta E., G.E. Cantarella A Day-to-Day and Within-Day Dynamic Stochastic Assignment Model *Transp. Res. 25A pp.277-291* **1991**

Di Gangi M. Continous Flow Approach in Dynamic Network Loading *presented at the 2nd Capri Seminar on Urban Traffic Networks* **1992**

Drissi-Kaïtouni O. A. Hameda-Benchekroun, A Dynamic Traffic Assignment Model and a Solution Algorithm *Transp. Sci. 26 pp. 119-128* **1992**

Friesz T.L., F.J. Luque, R.L. Tobin, B.W. Wie Dynamic User Optimal Traffic Assignment Model Based on Optimal Control Approach *Op. Res. 37 pp.893-901* **1989**

Hamerslag R. Dynamic Assignment in Three-Dimensional Time Space *Transp Res. Rec. 1220 pp.28-32* **1988**

Ho J.K. A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem *Transp. Sci. 14 pp.295-305* **1980**

Ho J.K. Solving the Dynamic Traffic Assignment Problem on a Hypercube Multicomputer *Transp. Res. 24B pp.443-451* **1990**

Janson B.N. Dynamic Traffic Assignment for Urban Road Networks *Transp. Res. 25B pp.143-161* **1991**

Mahmassani H.S., G.L. Chang Experiments with Departure Time Choice Dynamics of Urban Commuters *Transp. Res 20B pp.297-320* **1986**

Mahmassani H.S., G.L. Chang Travel Time Prediction and Departure Time Adjustment Behaviour Dynamics in a Congested Traffic System *Transp. Res 22B pp.217-232* **1988**

Merchant D.K., G.L. Nemhauser A Model and Algorithm for the Dynamic Traffic Assignment Problem *Transp. Sci. 12 pp. 183-199* **1978a**

Merchant D.K., G.L. Nemhauser Optimality Conditions for a Dynamic Traffic Assignment Model *Transp. Sci. 12 pp. 200-207* **1978b**

Papageorgiu M. Dynamic Modelling, Assignment and Route Guidance in Traffic Networks *Transp. Res. 24B pp.471-495* **1990**

Ran B., T. Shimazaki A General Model and Algorithm for the Dynamic Traffic Assignment Problem *Proceedings of the 5th World Conference on Transportation Resarch - Yokohama, Japan* **1989**

Ran B., D.E. Boyce, L.J. LeBlanc Dynamic User-Optimal Departure Time and Route Choice Model: A Bilevel, Optimal Control Formulation *Submitted to Ann. of Op. Res., special issue on Advances in Equilibrium Modelling, Analysis and Computation* **1992a**

Ran B., D.E. Boyce, L.J. LeBlanc Dynamic User-Optimal Route Choice Models Based on Stochastic Route Travel Times *presented at the 2nd Capri Seminar on Urban Traffic Networks* **1992b**

Smith M. J. A New Dynamic Traffic Model and the Existence and Calculation of Dynamic User Equilibria on Congested Capacity-constrained Road Networks *Transp. Res. 27B pp 49-64* **1993**

Vythoulkas P.K. A Dynamic Stochastic Assignment Model for the Analysis of General Networks *Transp. Res. 24B pp.453-469* **1990**

Wie B.W., T.L. Friesz, R.L. Tobin Dynamic User Optimal Traffic Assignment on Congested Multidestination Networks *Transp. Res. 24B pp.431-442* **1990**

Yagar S. Emulation of Dynamic Equilibrium in Traffic Networks *Traffic Equilibrium Methods pp. 240-264 (Florian M. Editor) Springer-Verlag* **1976**

On a Game Theory Approach in Transportation Systems Analysis

Alexander S. Belenky

Institute of Control Sciences, Russia Academy of Sciences

In many situations related to trade and transportation systems analysis one may face with necessity of modeling a bargain participants behavior when at least 3 participants act and the interests of at least two of them do not coincide.In such situations it is convenient to consider the participants as players in certain games and one of the most important problems from both theoretical and practical view points is to analyze whether an equilibrium point in such games exist and how to calculate it if so. Some advantages and disadvantages of such an approach to the modeling of the mentioned above situations are considered.

A class of antagonistic games on polyhedral sets where two of three bargain participants (buyer, seller and carrier) have the same interests and act as a unified player was considered by the author and discussed, in particular, at TRISTAN in 1991. In this paper we mostly explore a class of 3-person games on polyhedral sets with some kinds of nonlinear payoff functions. Some practical situations which can be modeled in such a form are discussed. Verified necessary and sufficient conditions for a Nash point in considered games are established and a finite method for its calculation based on those conditions is proposed. It turns out that one can calculate vector-components of the Nash point by solving some auxiliary linear and quadratic programming problems formulated on the basis of the master game.

Among the other problems under consideration in the paper are 2-person games on polyhedral connected sets and 4-person games on polyhedral sets. Some well known practical situations appeared in transportation systems analysis for which those problems can serve as their models are described and approaches to the solution methods construction for the problems are discussed.

On-Line Searching of Graphs and Lattices[*]

Patrick Jaillet[‡]

1 Introduction and Motivations

There are many situations in which present actions must be made and resources allocated with incomplete knowledge of the future. The difficulty in these situations is that we have to make our decision based only on the past and the current task we have to perform. It is not even clear how to measure the quality of a proposed decision strategy. The approach usually taken is to devise some probabilistic model of the future and act on this basis. This is the starting point of the theory of Markov Decision Processes (see for example [7]). The approach, around which this paper is based, is to compare the performance of a strategy that operates with no knowledge of the future (on-line) with the performance of an optimal strategy that has complete knowledge of the future (off-line). This measure takes the following approach in analyzing the performance of an on-line algorithm: An on-line algorithm is good only if its performance on *any* sequence of requests is within some (desired) factor of the performance of the off-line algorithm.

In this paper we are concerned with the problem of finding shortest paths in unknown environments, and with the development of deterministic and randomized on-line algorithms for this very general class of problems. More precisely, we consider the following general problem:

On-Line Graph Searching problem: Given a graph $G = (V, E)$ with nonnegative weights on its edges, and two distinguished nodes s and t, a searcher has to start from node s and find node t. We assume that, at any point of his search, the searcher's knowledge of the graph consists of all the nodes visited so far, of the weight of the edges incident to these nodes, and of the nodes adjacent to the visiting nodes. We also assume that the searcher recognizes the exit only when directly visiting

[*] Extended abstract submitted to *TRISTAN II*

[‡] MSIS Department, The University of Texas at Austin, and Laboratoire de Mathématiques Appliquées, ENPC, France. Supported in part by a joint NSF/CNRS research grant and by a Faculty research grant of The University of Texas at Austin

it. An edge can be traversed in either direction, but its weight is added to the total distance at each traversal. The problem is to find a strategy that minimizes the worst case ratio between the total distance traveled and the length of the shortest path from s to t.

Three special cases of this On-Line Graph Searching problem are analyzed in details: The "m-Disjoint Path" problem, for which the graph G contains m paths which, with the exception of a common source s, are vertex disjoint; The "Lattice" problem, for which the underlying graph is the natural infinite grid; and The "m-Regular Tree" problem, for which the graph is a rooted regular balanced tree of outdegree (i.e., number of children) m, and for which the source node s is the root of the tree.

In addition to finding optimal or near optimal on-line algorithms for these problems we consider the following basic questions: (i) What is the "value" of getting additional information before starting the search? and (ii) What is the relative power of having several searchers in parallel as opposed to having a single searcher with additional initial information?

1.1 Motivations

The On-Line Graph Searching problem is central to many areas of Computer Science, Operations Research, and Transportation Science. One obvious application is in the area of robotics, where such problems come up repeatedly whenever a robot, exploring an unknown environment, faces an obstacle and tries to find the best way to avoid it. Also, this problem is a simple and stylized version of general problems for which a target or a boundary has to be located by successive moves in a largely unknown search space (see [2, 4, 5, 6, 8]). Moreover, In addition, the problem of On-Line Graph Searching generalizes numerous other on-line problems, and, for this reason, is a key problem in the analysis of on-line strategies. In fact, as pointed out in [3], the special case of traversing layered graphs (see Section 1.4) is by itself a generalization of the Metrical Task Systems problem and K-server problem.

This body of research also serves as an important paradigm in decision-making under incomplete information, and as such has numerous other practical applications. Finally, "How much is it worth to have additional information before solving a problem?" is one of the major theoretical and practical motivation behind our line of research.

1.2 Previous Work and Related Problems

Baeza-Yates, Culberson, and Rawlins [1] discuss strategies for the m-Disjoint Path problem when the weights of the edges are all equal to one, hereafter denoted the "Equal Weight m-Disjoint Path" problem. They propose and prove that the following strategy has an optimal competitive ratio: move in the integrally increasing powers of $m/(m-1)$ in a round-robin manner, visiting paths in the same order over and over again. This strategy yields a competitive ratio $r_m = 1+2m(1+\frac{1}{(m-1)})^{m-1} \sim 2em$ as a function of the number of lines m.

A different but related problem has also recently appeared in the literature under the name of the "Layered Graph Traversal" problem. A *layered graph* is a connected weighted graph whose nodes are partitioned into sets (i.e., layers) $L_0 = \{s\}, L_1, L_2, \ldots$ and all edges connect nodes in consecutive layers. The edges between layer L_i and layer L_{i+1} are *all revealed* when the searcher visits some nodes in L_i (this is the main difference with the On-Line Graph Searching problem). This problem is introduced in Papadimitriou and Yannakakis [11] and is solved optimally for the case of m disjoint paths, by using the results of [1]. More recently Fiat, Foster, Karloff, Rabani, Ravid, and Vishwanathan [3] give upper and lower bounds on the best competitive ratio of layered graph traversal algorithms in the general case, and Ramesh [12] improve substantially on these bounds.

2 Outline of some results

In the first part, we first consider the Equal Weight m-Disjoint Path problem and assume that the searcher knows that the exit is within n steps from the source. For $m = 2$, we develop and prove the optimality of a family of strategies that depend on n. When n goes to infinity, this approach provides an alternative optimal strategy to the one proposed in [1]. However, for any finite n (even very large), it is shown to be much better than this last one (see Section ??). A basic idea of the approach is to solve the following dual question: Given a competitive ratio r, what is the largest "extent" (i.e. the farthest we go in all direction) of the graph that can be searched without violating this ratio? This idea is then generalized to $m > 2$. These results can easily be extended to cases where the additional information is either: the exit is at a distance more than l from the source; or: the exit is at a distance more than l and less than n ($n > l$) from the source. Also, we have investigated the case of the Equal Weight 2-Disjoint Path search in which node s is at 0 and node t is assumed to be equally likely on $[-n..n] \setminus \{0\}$. In such a context, we have numerically found the best strategies, and have shown that when n goes to infinity the strategy of [1] is still good, but not best. Finally we can easily extend the analysis to the problem with weights 0 or 1, and then to general nonnegative weights. We then extend all the results to searches on lattices.

In a second part, instead of getting additional information on the position of the exit to be found, we consider additional searchers that would search the exit in a parallel fashion. In that case, the Graph Searching problem can have many different flavors, depending on: (a) The level of communication between searchers (either total communication or communication only when the exit is located), (b) The definition of the "cost" of a strategy (either time elapsed or total distance covered), and (c) The problem itself (either locate the exit or have all searchers go to the exit). For example, we can easily prove that, in the case of searching m disjoint paths with k communicating searchers ($k < m$), and if we want to minimize the total distance traveled, the optimal strategy, among strategies that partition the tasks among searchers, is to assign one path to each of $k-1$ searchers, and assign the remaining $m-k+1$ to the last searcher! With the other cost measure (time elapsed), a more balanced load is obtained.

In a third part, we investigate the "m-Regular Tree" problem and concentrate on the case for which the weights of the edges are all one, denoted the "Equal Weight m-Regular Tree" problem. The best proved strategies correspond to a family of algorithms (indexed by m) which are based on a careful combination of "alternating depth-first search" strategy. This strategy starts at the root of the tree, then visits, using depth-first search, part of each of the m subtrees, in a round-robin manner, incrementally increasing the depth of the subtree to be searched, and doing it over and over again until the bottom of the tree is reached or the exit is found. We then extend our result to 0-1 and then general weights. Finally any general tree which has node degrees bounded by d and diameter D can be searched without violating a competitive ratio $r(d-1,h)$, which is the best competitive ratio for a rooted regular balanced tree of outdegree $d-1$ with height $h = D$. This last problem is the $d-1$-Regular Tree problem. One can then use the previous on-line algorithm in an adaptive fashion: We start at the given node s, we search the tree assuming we have a regular balanced tree of outdegree d^0, where d^0 is the degree of s. We then update d while visiting the tree and finding nodes of higher degree.

References

[1] R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and*

Computation, (to appear), 1991.

[2] R. Bellman. A minimization problem. *Bulletin of the American Mathematical Society*, 62:270, 1956.

[3] A. Fiat, D. Foster, H. Karloff, Y. Rabani, Y. Ravid, and S. Vishwanathan. Competitive algorithms for layered graph traversal. In *Proc. of the 32nd Ann. IEEE Symp. on Foundations of Computer Science*, pages 288–297, 1991.

[4] B. Gluss. An alternative solution to the 'lost at sea' problem. *Naval Research Logistics Quaterly*, 8:117–128, 1961.

[5] B. Gluss. The minimax path in a search for a circle in the plane. *Naval Research Logistics Quaterly*, 8:357–360, 1961.

[6] R. Hassin and A. Tamir. Minimal length curves that are not embeddable in an open planar set: The problem of a lost swimmer with a compass. *SIAM Journal of Control and Optimization*, 30:695–703, 1992.

[7] D. Heyman and M. Sobel. *Stochastic Models in Operations Research*, volume 2. Mc Graw-Hill, 1984.

[8] J. Isbell. An optimal search pattern. *Naval Research Logistics Quaterly*, 4:357–359, 1957.

[9] M. Kao, J. Reif, and S. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. In *Proc. of the 4th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 441–447, 1993.

[10] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator. Competitive snoopy caching. *Algorithmica*, 3:79–119, 1988.

[11] C. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84:127–150, 1991.

[12] H. Ramesh. On traversing layered graphs on-line. In *Proc. of the 4th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 412–421, 1993.

[13] D. Sleator and R. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

Computer Tool for Designing and Analysis of Transport Networks (WTRANS)

Purtov A.M., Tokarev Yu.P., Shaptsev V.A., Shulman V.B.

Introduction

The next problems of motor transport are the most important for both Omsk and other large cities in Russia: raising for traffic safety; efficient (from the point of view of decrease in waiting time on the stops) organization of passenger transport activity; minimization of transport expenses while transportations. These problems and some other one require immediate solutions of numerous tasks that arised, generally, owing to spontaneous in its basis development of transport network of the city. Some of them are specific for Russia: elaboration of transport development conception, raising of roads traffic capacity, construction of new motorways, viaducts, tunnels, transport flyovers, improvement of the road covering quality. Other problems are general one for developing cities: development of transport traffic layot, distribution of transport streams, creation of route network of passenger transportations, optimization of transportations, scheduling of transport facilities activity, placing of filling stations, service stations, etc.

The elaboration of these problems on a system layer and with application of the methods of optimal solution search require different mathematical models and methods from the field of combinatorial optimization, schedules theory, geometric programming and other branches of mathematics. Many problems require nonformal using of knowledges and intuition of experts. Powerful combination of all this arsenal can be reached by means of up-to-date information technologies that include means of models visualization and results of computer experiments with them. Therefore, when using a personal computer for transport problems solution they give much attention for the quality of graphics representation of the investigated objects.

Transport systems objects are characterized by high dynamic that increases the complexity of their representation on the screen. It is very hard to create transport systems icons and their elements. On the other hand, while solving different transport problems they often use standard objects, such as graphs, transactions, servers, queues. So it'll be usefull to create a system that includes the set of standard objects intended for transport problem solutions on a high layer of abstraction.

The software market in the field of transport problems solution has quite perfect products. But the states of former Soviet Union can't afford to buy such software both due to their high cost and lack of appropriate class of PCs. Therefore, applied software specialists have to work at the projects that run on available PCs and very often the solved problems have not practical value. Nevertheless, by means of these experiments program specialists keep their professional skills and they are able to apprehend up-to-date technologies at the moment of their appearance

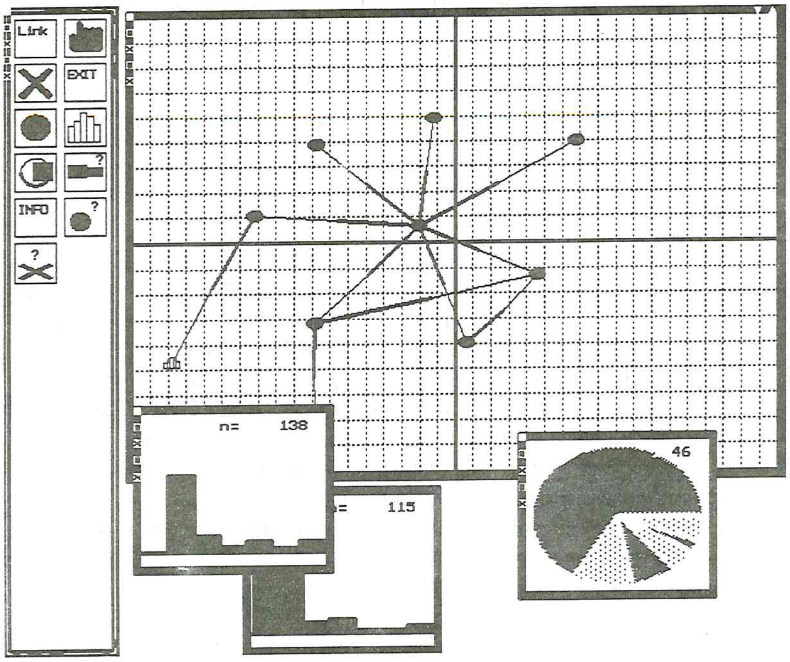


Figure 1: *Interface example*

in Russia. At the same time transport specialists are being trained as potential consumers of perspective design technologies and development management.

Description of WTRANS System

The following WTRANS system serves for solution of characterisatics estimation problems, optimization, transport network simulation operation represented as graphs and/or queueing system networks. WTRANS takes the second layer in the hierarchy of the computer tools for complex systems modelling created in IITAM. Universal system of simulation and representation of discrete objects (W) is on the first layer [1]. Curently, this tool of models designing and investigation is being used. The part of its interface is represented on Fig.1.

On the third layer there are problem-oriented modelling tools which allow to investigate model objects, ex.simulation networks tools (WNETWORK). All layers are given in C++ language using object-oriented approach which allows to inherit low levels transport systems objects useful characteristics of those of high levels. To understand the characteristics of WTRANS system objects, it's necessary to describe basic possibilities of W system.

This system offers its users the following, general possibilities: discrete event systems tools which are analogues to the language SIMULA-67; tools for icons manipulation on the display screen (creation, scaling, moving, copying, etc.); statistic data collection and representation tools; standard objects of frequent use (graph node, graph edge, etc.); object menu creation tools; tools for conversation with a model and modelling process organization; tools for creation of modelling system in a specific domain.

All above-mentioned possibilities of W system are used in WTRANS creation. WTRANS consists of the following basic objects standard for transport problems and queueing systems: transactions; transaction source; transport network node; transport network link if transport connection between nodes is available; route; router.

Transactions are objects that move on transport network and simulate movement of products in space (goods, passengers, data). Transactions parameters are: transaction sender-node, transaction receiver-node; transit node; quantity of transported products V; the moment of creation of transaction.

Transactions source simulates the receiving of requests for transportation and generates transactions in accordance with the given law. Transactions source has the following parameters: transaction sender node; transaction receiver node; parameters that issue distribution of random value V; parameters that issue distribution of time intervals between transaction generation.

Basic functions of transport network node are: transactions generation and termination; routing; transmission of transactions in the link and receiving them from the link. Extra functions may be added to the node while solving definite problem which should be programmed beforehand and be included into the structure of node methods.

Transport network link serves for modelling of transactions transfer between nodes. Several links may exist between couple of links in each direction. The link has the following parameters: transmitter node; receiver node; capacity; parameters that issueg the spread of time of transaction transfer on the link; cost of transaction transfer on the line C. Parameter C serves for route optimization.

Parameter C can be represented, for example, by average time of transaction transfer on the link; current length queue to the link or any other criterion which is chosen by the user while describing router.

Nodes and links coordinates are assigned on the display screen by the mouse. Besides, they may transfer themselves and terminate themselves. Transport network may be found on the screen simultaneously in several windows in different scales. The change of configuration in a window automatically changes the other one.

Route is a sequence of links which transaction passes from sender to receiver. It can be calculated by router or it can be indicated by the user with the help of the mouse during model generation.

Router is a set of methods to choose routes is open for enhancement. Nowadays method of adaptive routing is work out. It transmits transactions on the shortest route from the point of view of given criterion C. The length of the route is calculated as the sum of parameters C values for the links which are included in this route. This method requires following parameters: step of the routes modification; number of the given cost criterion N. When $N = 0$ all routes are given by the user setting input data. When $N = 1$ parameter C value is 1 for all links route length equals the number of the links. In this case transactions are transmitted on the shortest routes from the point of view of the number of transit node. When $N = 2$ the value C is calculated as the average time of transaction waiting in queue to the link. And average time is calculated according to the set of transactions transmitted on the link from the moment of the last route modification. In this case adaptive routing is carried out. Transactions are transmitted by the shortest routes from the point of view of their waiting in queue.

It is not difficult to change the calculating algorithm for parameter C, so the length of the route set can be easily added. The problem of analyzing the efficiency of different criteria for route length and the problem of the choosing criterion C which optimizes the efficiency of the transport network given index have great scientific and practical interest.

In conclusion of the review oi WTRANS system basic objects it should be said that this set is open for complementation by other objects. Besides there are some support statistic objects as: queue; histogram; table. Statistic objects are joined to the transport network main objects when setting input data or in the process of model execution and serve for statistic data processing and representation (see Fig.1)

Libraries of Classes of WTRANS System

Domain classes library is an intellectual basis of the given tool. Microsoft Foundation Classes' library is a good example for creation of specialized classes library. But the cost of the appropriate libraries is too high, at least in Russia. Therefore, in our country they usually create their own classes libraries in each organization. The best you can expect when extending the possibilities of your tool is libraries exchange and their joint development by several elaborator teams.

Basis W system has the following classes libraris [1]: multiwindow graphics; discrete system modelling; creation of graphics information-search systems. The following libraries are created for modelling of network structures [1]: classes of package transmission networks with random multiple access (ALOHA, CSMA); classes of high layers of seven-layer model of open systems. All above-mentioned classes libraries are applied for solution of many transport problems. It is supposed to create new specialized libraries. In spite of this, the elaborators suppose the availability of some classes libraries on the market, such as [2]:

- user's graphics interfaces and programming under Windows: windows, menus, conversational windows, processing forms and functions (ex., Win++ —Blaise, C++/Views —Liant, XVT —XVT Software, zApps —Inmark);
- data types and sets classes (Booch Components —Rational, Microsoft Foundation Classes —Microsoft, Tools.h++ —RogueWave Software);
- interfaces of database management system that provid gateways with traditional relational database (Object Manager, db-VISTA III —Raima, POET —BKS, Data Object Manager —Code Farms);
- input-output streams (Unix System Laboratories' libraries);
- classes for programming of files transmission protocols, terminals emulation, Hayes-modem management, etc. (product Comm++ —Greenleaf Software/Comm++);
- classes for view and processing of numeric information: complex numbers, statistics, linear algebra, matrixes, etc. (complex —USL, Math.h++, Linpack.h++ —RogueWave Software, M++ —Dyad);

- Modelling and simulation (Meijin++, ObjectTime —Network Integrated Services);
- libraries for informational protection in bank activity (Infinity International Financial and Berkley Investment Technologies).

Conclusion

Approbation of WTRANS system has shown the following positive qualities: convenient user interface with the problem on the layer of the visual representation of the objects investigated which is useful when issueing input data and simulation experiments; sufficient generality for solution of many transport problems; good possibilities for new transport problems adaptation.

We should note some other elaborations which are being investigated in the Institute of Information Technologies and Applied Mathematics of SB RAS and intended for use of mathematical modelling as a method of transport problems solution. In particular, we develop algorithms and programs for problems solution of optimal conveying [3], decomposition algorithms for solution of industrial-transport problems of discrete type [4], program system for various problems solution on a transport network. The last one has several layers of source information which is used when solving some problems (topographic basis, transport network streets, street characteristics, railroads and their intersection with motorways, crossroads, turns, infrastructure objects, public passenger transport routes and stops, etc.). The oriented graph is the basis of mathematical model of transport network. There is a possibility of graph's superposition on a country and its simultaneous viewing with any other layer of the map, including other linear objects. Except merely informational function, this software allows to solve traditional network problems: commercial traveller, shortest way, contiguity and nearness, placing of resources and so on.

References

[1] V.A.Shaptsev, V.B.Shulman, Yu.P.Tokarev, The Computer Station for the Analysis of Hierarchical Systems, "3rd IFIP WG 7.6 Working Conf. on Optimization-Based Computer-Aided Modelling and Design", Prague, May 24–26, 1994, 7p. (in the press).

[2] IDC: Object technologies, Computer World, 1994, No 15 (123), pp.1, 28–29.

[3] A.A.Kolokolov, N.V.Kolmichevskaya, V.V.Servakh,V.V.Tsepkova, Dialogovaya sistema "Kontur" dlya optimizatsii grafikov dostavki khleboproduktov. Tezisi dokladov XI vsesoyuznoy shkoli po sistemam programmnogo obespecheniya resheniya zadach optimal'nogo planirovaniya. - Moscow, 1990. - p.159.

[4] A.A.Kolokolov, Decomposition algorithms for solving of some production-transportation problems. Abstracts of the Conference TRISTAN 2, Capri, 1994.

MULTICRITERIA EVALUATION MODEL
OF PUBLIC TRANSPORT NETWORKS

by

Maurizio Bielli and Massimo Gastaldi, IASI-CNR,
Viale Manzoni 30, 00185 Rome, ITALY

Pasquale Carotenuto, PFT2-CNR,
Viale dell'Università 11, 00185 Rome, ITALY

1. INTRODUCTION

Since the life quality in urban centers strictly depends on the efficiency of urban public transit, this topic has recently been the subject of some studies. The demographic and urban growth with the related increase of industrial, commercial and service activities, determine a growing mobility demand not supported by well-organized public transit supply. In fact, old networks have been updated without considering the system complexity, by adding and overlapping new links to the old ones not taking into account the high cost of utilized human and physical resources and the system functionality for the users. Thus, the necessity to support policy makers in choosing optimal decisions in order to plan and increase urban public transit is one of the main aims in the management of urban system. In fact, the interest of researchers is now to collect information on the planning problem, to define the various choice strategies which may be adopted, to specify in an operational way the planning objectives of decision-makers, to develop an evaluation procedure appropriate both for the decision problem and for the available information, to present optimal or at least reasonable solutions and to test the choice actually made by the decision-makers with respect to partial outcomes of the plan during the process of implementation.

In this paper a multicriteria evaluation methodology is applied on an Italian real network and the obtained results are presented. This analysis has been performed on the urban network of Parma, assuming as reference situation the existing public transit network and proposing three alternative networks. The first one is planned with respect to the demand whereas the remaining two alternatives depend on the physical characteristic of the proposed networks.

2. PROJECT EVALUATION

To design an efficient transportation system on the basis of possible alternatives, it is necessary to individuate the following steps:

1) Objectives individuation: reorganization of urban bus transit network with high values of performance, evaluated considering appropriate performance indicators. In this way the demand is static, in the sense that its variations could be considered negligible both during the required time for the application of the project and for the repercussions on the system (growing demand due to the project realization, O-D changes, etc.).
2) Analysis of existing situation: survey and evaluation of present network called network 0.
3) Definition of alternative project solutions: formulation of realizable alternative projects able to replace the present one (network 1, network 2, network 3).
4) Definition of the mathematical model representing the studied transportation system.
5) Evaluation of alternative projects: simulation of the transportation system on the basis of the available alternatives.
6) Optimal project selection: comparison among alternative projects through the evaluation of indicators that best describe the overall level of performance of transit bus system in order to select the optimal project and so the optimal network. For this purpose, a computer package EVA (Drive, Project V1036, 1991) has been used.

3. SIMULATION PROCEDURE

Bus transportation demand is given by the number of users utilizing the bus service in a prefixed time interval. Since we must verify the characteristics of the networks with the higher level of demand, the peak-hour interval has been considered. The transit bus network has been defined considering the following three phases: area delimitation, area division (zones) and graph extraction.

In the first phase, we delimit the geographical area containing the effects produced by the available projects aiming at reorganizing the urban bus transit system. Thus, we considered as area of study the city center and the immediate suburbs. The external areas have been considered only for their interconnections with the studied area. The aim of the second phase is that of identifying with a point, called centroid, the starting and ending points of pedestrian movements in a sub-area (zone). We delimit 100 zones in accordance with the census area of Parma facilitating the sample extraction for the demand estimation. In the last phase, we determine the significative positions of users for the definition of graph nodes and links. The nodes can represent a real physical point in the area of study (i.e. crossing point) or the condition of the user in the transportation system (i.e. step arrival, the time instant when the user really utilizes the bus service), and the links can represent an effective connection or a wait time.

Now, it is possible to define the alternative networks (projects). The first one (network 1) is demand-oriented; it considers the user needs utilizing an appropriate O-D matrix and creating suitable paths following the required movements. The second and the third network are developed considering the physical characteristics of the proposed networks: in the former we used a circular scheme completed with a set of crossing lines assuring the existence of overlapping points for interconnections. The latter used a scheme with half and radial rings.

Once the service supply has been defined, it is necessary to describe the path choice model. We utilized a behavioural model; the user choices occur in two different time instants: before leaving and during the travel. In the first case, the user utilizes his information on alternative paths (consuntive choice), whereas in the second case he can variate the first choice during the travel in accordance with additional information or with contingent events (adaptive choice). The former choice is based on a compensatory model: the user evaluates the attribute utilities of available alternatives in order to perform his choice. The latter is based on casual events during the travel; in this way the utilized path can be considered as an aleatory variable.

Moreover, two hypotheses have been assumed:

a) since the time cumulated distance of the lines useful to the user is lower than 12 minutes, we can assume the casuality of the user stop arrival instant.

b) the casuality of the vehicle stop arrival instant.

Considering these two hypotheses, the adaptive choice will be referred only to the choice of the vehicle on a line belonging to a prefixed user useful set.

4. PERFORMANCE INDICATORS

Evaluation represents an integral part of service management with respect to the design of new services or the improvement of an existing one. Thus, to provide an efficient performance and quality evaluation of traffic/transport services, it is necessary to consider many other typical social, political and geographical factors in addition to the basic technological-economic factors.

A large body of literature has emerged dealing with a wide set of measures (indexes, variations, ratios, etc.) of performance for transportation services (Henderson, Kwong and Adkins, 1991; Bielli, Gastaldi and Sica, 1992). This great availability of performance measures was followed by a number of research projects designed to reduce the set of measures to a manageable size. There is a general agreement that main impacts fall into three categories: efficiency, effectiveness and quality.

In classic economic terms, a production process is efficient if it is not possible to augment any output without increasing any input or if it is not possible to decrease any input without augmenting any other input to obtain a prefixed output. However, a great number of efficiency measures appears to be inappropriate for choosing among transportation options. A further complication is introduced by an ambiguity relating to the definition of system output. In fact, in transportation service perspective, output can be measured in terms of the amount of services available (e.g. vehicle-km., vehicle-hours, etc.) or in terms of the amount of services used (e.g. passenger-km. etc.). In general, efficiency measures indicate the performance of the system in producing service at minimum cost and are related to quantities of output produced. Cost per vehicle-hour is a basic measure of the efficiency of a service provided, which reflects both the cost of inputs such as labor and fuel and the quantity of inputs used to produce the service. Another typical indicator of transit system efficiency is the vehicle-hours per employee. The number and cost of accidents are also an efficiency measure because accidents represent an unproductive use of resources.

The effectiveness performance indicators measure service (the proportion of the service area's households that have access to the system). The percentage of households served is a basic indication of the coverage of the service area provided by a transit system. In fact, it typically serves most but not all the households in the political jurisdiction or service area within which it operates, depending on the availability and density of transit stops. For these reasons, measures normally used to test system efficiency combine efficiency with effectiveness elements; in this way, it is possible to analyze how the system really serves the users besides to investigate how the system utilizes input to produce service. Effectiveness measures indicate how well a transit system works and relate the quantity of output consumed with respect to the number of passengers. In particular, the cost per passenger indicates the average total operating cost for providing service to a passenger and the revenue per passenger is calculated by dividing system revenue by total passengers carried. From the passenger's point of view, the most effective system is one that requires no transfers. From the system's perspective, it is not possible to tailor the service so that everyone gets a one-vehicle ride from origin to destination.

Methodology for measurement of quality in the service sector is a controversial topic. Quality evaluation for an urban transportation system means comparing the considered service level with the competitor ones.

Moreover, urban and metropolitan areas are very complex and decision making is characterized by a large number of conflicting goals and objectives postulated by various groups; then it is necessary to take into account the decision maker perspective.

Table 1 shows the set of indicators utilized in numerical results to evaluate the available alternatives for the improvement of bus transit service in Parma. All of them are referred to the same time unit, the peak-hour, and calculated utilizing vehicle capacity (100 per vehicle), number of users per line, frequency, length of the line and number of vehicles per line. For the efficiency indicators we assume as output unit the vehicle capacity per km and the number of vehicle per km and as resources the line frequency and the minimum number of vehicle per line. For the effectiveness indicators we assume the number of users per km as output unit.

Performance indicators	**Symbol**		
Vehicle number	Vn	▼	E
Vehicle capacity per Km/Vehicle number	CKM/Vn	▲	E
Vehicle number/Frequency	Vn/F	▼	E
Users	U	▲	e
Users/Vehicle capacity per Km	U/CKM	▲	E
Users/Network extension	U/L	▲	e
Users/Vehicle number	U/Vn	▲	E
Users per Km	UKM	▲	e
Users per Km/Vehicle capacity per Km	UKM/CKM	▲	E
Users per Km/Network extension	UKM/L	▲	e
Users per Km/Vehicle number	UKM/Vn	▲	E
Number of stops	Ns		e
Number of stops/Number of links	Ns/Nlk	▲	e
Number of links	Nlk	▼	E
Number of lines	Nln	▼	E
Average number of transfers	Nt	▼	Q
Average time of wait	Tw	▼	Q
Average time of travel	Tt	▼	Q
Average time of moving	Tm	▼	Q
Number of pedestrian relations	Npr	▼	e
Number of pedestrian	Np	▼	e

Effectiveness indicators	E
Efficacy indicators	e
Quality indicators	Q

The best value is the lowest one	▼
The best value is the highest one	▲

Tab. 1 - Performance Indicators

5. MULTICRITERIA ANALYSIS AND NUMERICAL RESULTS

The problem of decision-making with multiple criteria has been the subject of many different sciences, i.e. operation research, decision theory or regional and transportation planning. For this reason, a lot of different evaluation and decision techniques that could be called multicriteria analysis exist. Common for all is that they are able to deal with two or more criteria measured in different units. Evaluation is basically a comparison of alternatives; this is a multistage process which encompasses the establishment of evaluation criteria, the determination of individual values and the weighted up of the different impacts. An evaluation framework is the set of evaluation tools and components used in the evaluation process. First of all the evaluation method may be fixed with the corresponding evaluation criteria and impact values. Within multicriteria decision analysis different criteria and impacts measured with monetary or non-monetary values are considered in aggregate or disaggregate form. Several different techniques can be applied according to the type of criteria or impacts and to their aggregation, as dominance analysis, compatibility analysis, cost-effectiveness and utility analysis.

In a Multicriteria evaluation, the alternatives are compared with respect to a set of criteria corresponding to the policy objectives represented by performance indicators.The first major step in the evaluation process is the definition of the impact matrix (Bobinger et al., 1991); the impacts e_{ij} related to the individual criteria c_j (j=1,2,......,J) and the individual alternatives a_i (i=1,2,......,I) then establish the impact matrix. A weight must be designed for each criterion reflecting the normative judgement of the decision-maker. Thus, the evaluation can be summarized in the following five steps:

1) Project alternatives individuation.
2) Objectives or criteria individuation.
3) Calculation of impacts matrix.
4) Transformation of impact measures in utility measures.
5) Optimal project selection.

The project alternatives are incompatible in the sense that the realization of each alternative precludes the realization of other ones. Notice that the decision maker is the urban transit agency; however, the above mentioned performance indicators take into account the users and community point of view.

A group of indicators has been defined for each criterion weighted on the basis of its importance. The relative impact matrix is shown in Table 2.

	Weights	Network 0	Network 1	Network 2	Network 3
Vn	10	64,000	51,000	61,000	56,000
CKM/Vn	10	749,510	765,980	816,540	740,600
Vn/F	5	0,570	0,540	0,560	0,510
U	10	21547,000	21569,000	22134,000	21085,000
U/CKM	5	0,450	0,550	0,440	0,510
U/L	5	131,820	162,030	137,430	146,920
U/Vn	10	336,670	422,920	362,850	376,520
UKM	8	44004,960	43537,680	42523,580	43200,480
UKM/CKM	10	0,920	1,110	0,850	1,040
UKM/L	5	269,210	327,070	264,020	301,020
UKM/Vn	8	687,580	853,680	697,110	771,440
Ns	0	459,000	459,000	459,000	459,000
Ns/Nlk	7	0,152	0,161	0,153	0,159
Nlk	7	3016,000	2843,000	3006,000	2892,000
Nln	5	22,000	18,000	20,000	20,000
Nt	8	0,690	0,740	0,910	0,690
Tw	9	651,920	717,680	790,320	740,540
Tt	7	1385,440	1457,890	1523,740	1471,050
Tm	10	2163,030	2213,710	2311,040	2258,480
Npr	8	3,000	2,000	3,000	2,000
Np	8	44,000	28,000	60,000	28,000

Tab. 2 - Criterion weights and impacts matrix

In order to perform the multicriteria analysis we should consider a value function for each criterion (performance indicator), transforming the initial impact in a score easier to interpret. For this purpose, we used a value function f_j in the following forms:

a) when utility grows while impact decreases (i.e. cost case):

$$u_{ij} = \frac{\min (e_{ij})}{e_{ij}}$$

for each alternative i related to impact j

b) when utility grows while impact increases (i.e. benefit case):

$$u_{ij} = \frac{e_{ij}}{\max (e_{ij})}$$

for each alternative i related to impact j

Table 3 shows the relative weights and the matrix of performance indicators re-evaluated through the utility functions.

By examining this matrix we should obtain the indications necessary to choose the best alternative. On this subject, the current literature proposes different approaches; in our numerical results, we apply the Utility Value Analysis (UVA) by using the software package EVA. The stages of a Quantitative Utility Value Analysis are presented in Figure 1.

The Utility Value Analysis (UVA), a Multicriteria Analysis technique which aggregates the multiple criteria, is here applied. For each alternative a_i (i=1,2,......,I) and criterion c_j (j=1,2,......,J) the criterion-related impact e_{ij} has to be determined. These impacts e_{ij} are measured in various units (i.e. money, time, etc.). In order to make them comparable they are firstly transformed into a single unit utilizing the value function f_j (j=1,2,......,J) determined for each individual criterion cj. These functions transform the initial impact e_{ij} into the so called criterion utilities u_{ij} interpreted as goal-achievement scores. The form of value function depends on the preferences of the decision-makers.

In a second stage the individual criteria are weighted assigning a weight w_j to each criterion c_j; these weights reflect the relative importance of the criterion (the weights sum is usually equal to one). Similarly to the value function, the weights assigned to the criteria depend on the preferences of the decision

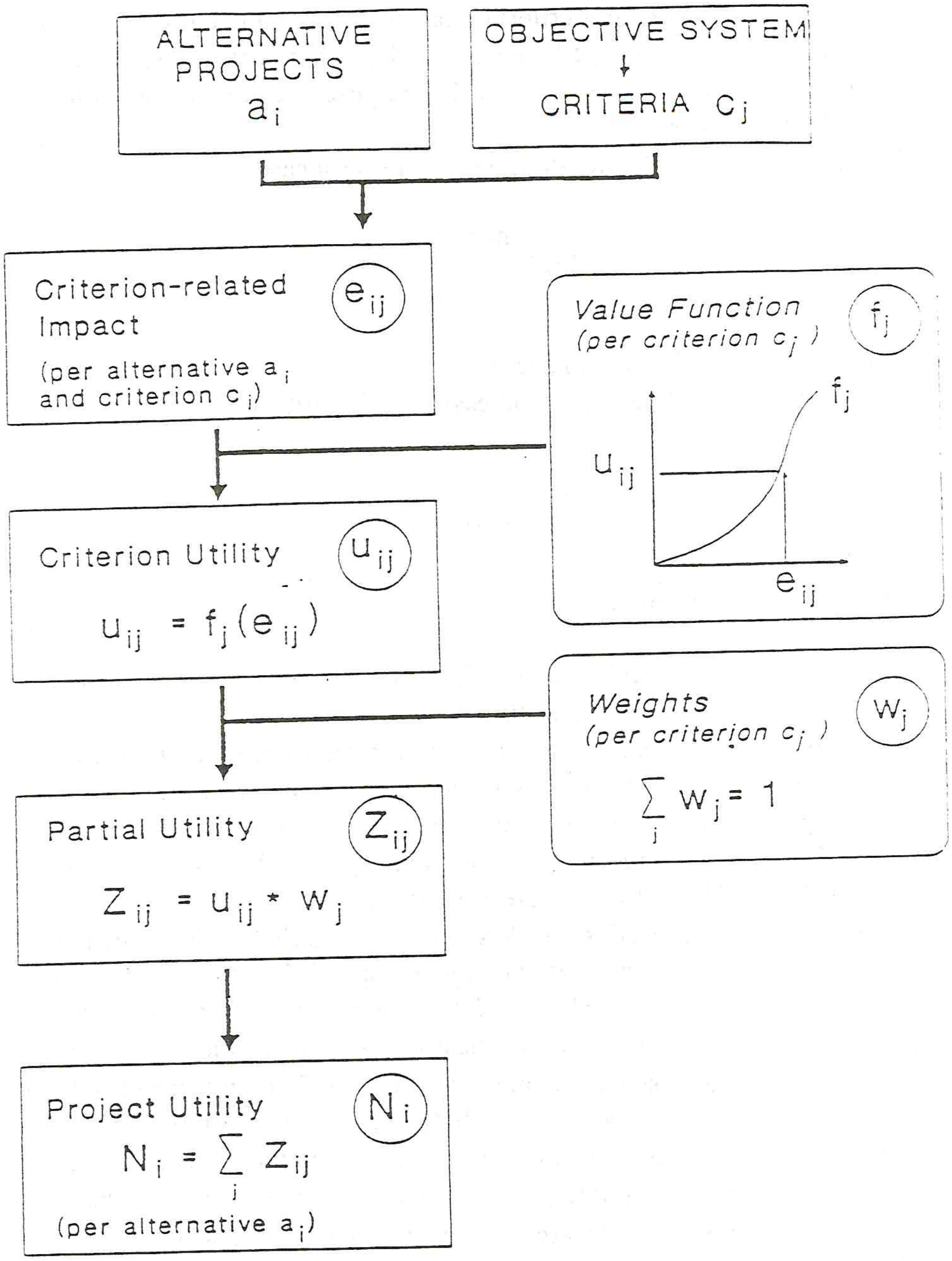


Fig. 1 - Stages of a Quntitative Utility value analysis

	Rel. weights	Network 0	Network 1	Network 2	Network 3
Vn	0,065	0,797	**1,000**	0,836	0,911
CKM/Vn	0,065	0,918	0,938	**1,000**	0,907
Vn/F	0,032	0,895	0,944	0,911	**1,000**
U	0,065	0,973	0,974	**1,000**	0,953
U/CKM	0,032	0,818	**1,000**	0,800	0,927
U/L	0,032	0,814	**1,000**	0,848	0,907
U/Vn	0,065	0,796	**1,000**	0,858	0,890
UKM	0,052	**1,000**	0,989	0,966	0,982
UKM/CKM	0,065	0,829	**1,000**	0,766	0,937
UKM/L	0,032	0,823	**1,000**	0,807	0,920
UKM/Vn	0,052	0,805	**1,000**	0,817	0,904
Ns	0,000	**1,000**	**1,000**	**1,000**	**1,000**
Ns/Nlk	0,045	0,944	**1,000**	0,950	0,988
Nlk	0,045	0,943	**1,000**	0,946	0,983
Nln	0,032	0,818	**1,000**	0,900	0,900
Nt	0,052	**1,000**	0,932	0,758	**1,000**
Tw	0,058	**1,000**	0,908	0,825	0,880
Tt	0,045	**1,000**	0,950	0,909	0,942
Tm	0,065	**1,000**	0,977	0,936	0,958
Npr	0,052	0,667	**1,000**	0,667	**1,000**
Np	0,052	0,636	**1,000**	0,467	**1,000**

Tab. 3 - Criterion relative weights and utilities makers. The weighted transformed criterion utility then reflects the partial utility (Z_{ij}).

Finally, these partial utilities have to be aggregated for each alternative. Due to the fact that preferential independence of the criteria are assumed, the aggregation can be performed by simple sum. Thus the overall utility (N_i) of alternative a_i is the sum of the weighted and transformed impacts. Where an alternative is superior to all others, no further analysis is required.

The above mentioned Tables 1-3 allow to follow the different stages to which performance indicators must undergo through the EVA package in order to highlight the best solution among the proposed ones. Table 4, showing the final indicator values, clearly indicates that the best alternative is Network 1, that is the demand-oriented Network presenting the higher aggregated utility index (fig. 2). Notice that two of the proposed alternatives are better than the existing one. Moreover, although some networks give the higher values for some specific criteria, they are not the best as a whole as shown in fig. 3.

The authors are currently working on comparing their results with other multicriteria techniques and on the possibility of adding new performance indicators to consider relevant topic in urban transport evaluation (i.e. pollutant emissions, urban trip quality, etc.)

REFERENCES

Bielli M., Gastaldi M. and Sica F. "Performance Evaluation in the Management of Urban Transportation Services", Proceedings of 1st Meeting of the EURO Working Group on Urban Traffic and Transportation, Landshut, Germany, 1992.

Bobinger R., Flowerdew R., Hammond T., Himanen A. and Keller H. "Context and framework of Drive transportation evaluation" in Proceedings of Drive Conference, Brussels, Elsevier, Amsterdam, 389-412, 1991.

DRIVE PROJECT V1036, EVA Manual - Evaluation Process for Road Transport Informatics, 1991.

Henderson G., Kwong P. and Adkins H. "Regularity indexes for evaluating transit performance", TRB Record 1297, 1991.

	Network 0	Network 1	Network 2	Network 3
Vn	0,051	0,065	0,054	0,059
CKM/Vn	0,059	0,061	0,065	0,059
Vn/F	0,029	0,030	0,029	0,032
U	0,063	0,063	0,065	0,061
U/CKM	0,026	0,032	0,026	0,030
U/L	0,026	0,032	0,027	0,029
U/Vn	0,051	0,065	0,055	0,057
UKM	0,052	0,051	0,050	0,051
UKM/CKM	0,053	0,065	0,049	0,060
UKM/L	0,027	0,032	0,026	0,030
UKM/Vn	0,042	0,052	0,042	0,047
Ns	0,000	0,000	0,000	0,000
Ns/Nlk	0,043	0,045	0,043	0,045
Nlk	0,043	0,045	0,043	0,044
Nln	0,026	0,032	0,029	0,029
Nt	0,052	0,048	0,039	0,052
Tw	0,058	0,053	0,048	0,051
Tt	0,045	0,043	0,041	0,043
Tm	0,065	0,063	0,060	0,062
Npr	0,034	0,052	0,034	0,052
Np	0,033	0,052	0,024	0,052
	0,878	**0,979**	**0,850**	**0,943**

Tab. 4 - Partial and project utilities

Figure 2 - Project utilities

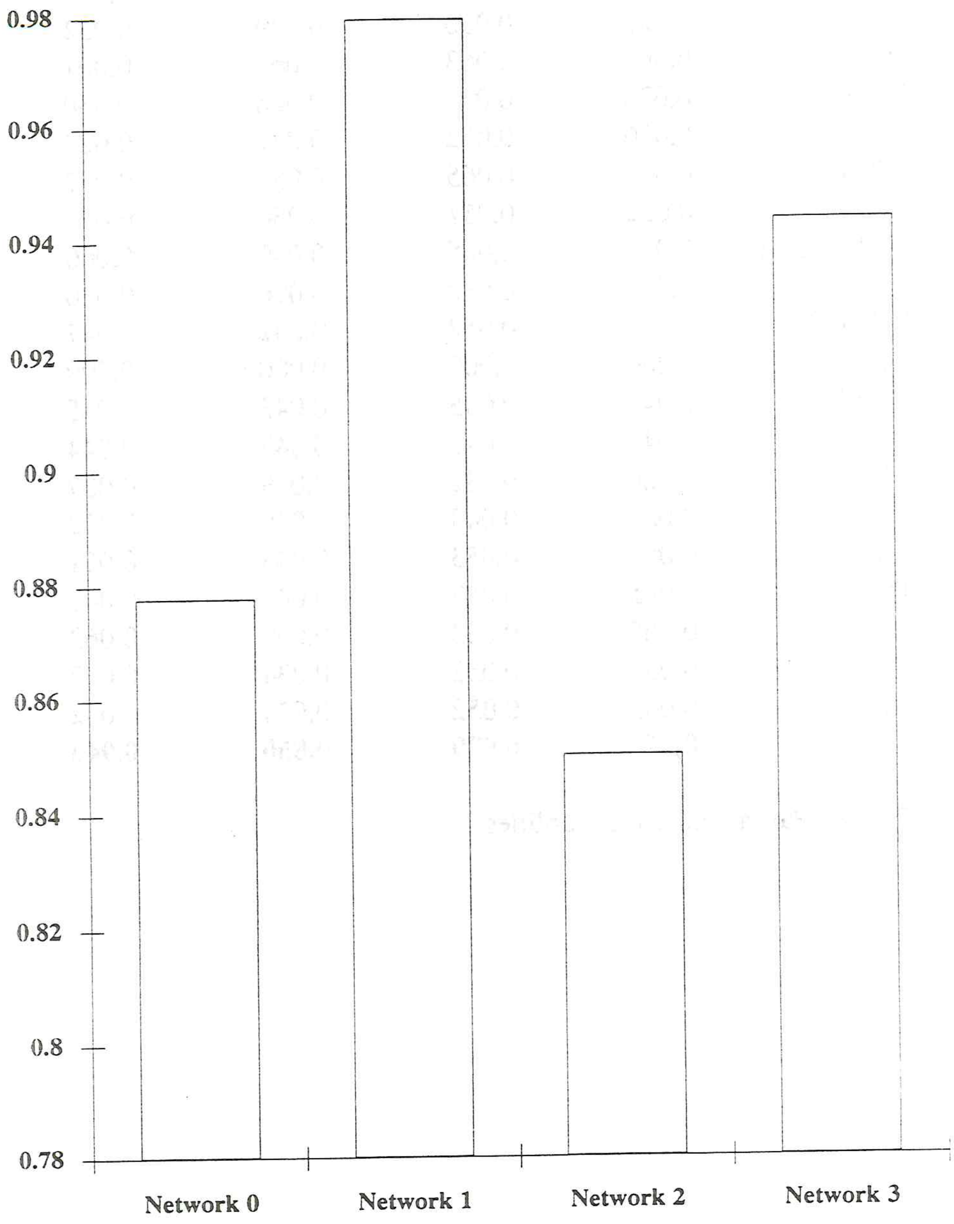
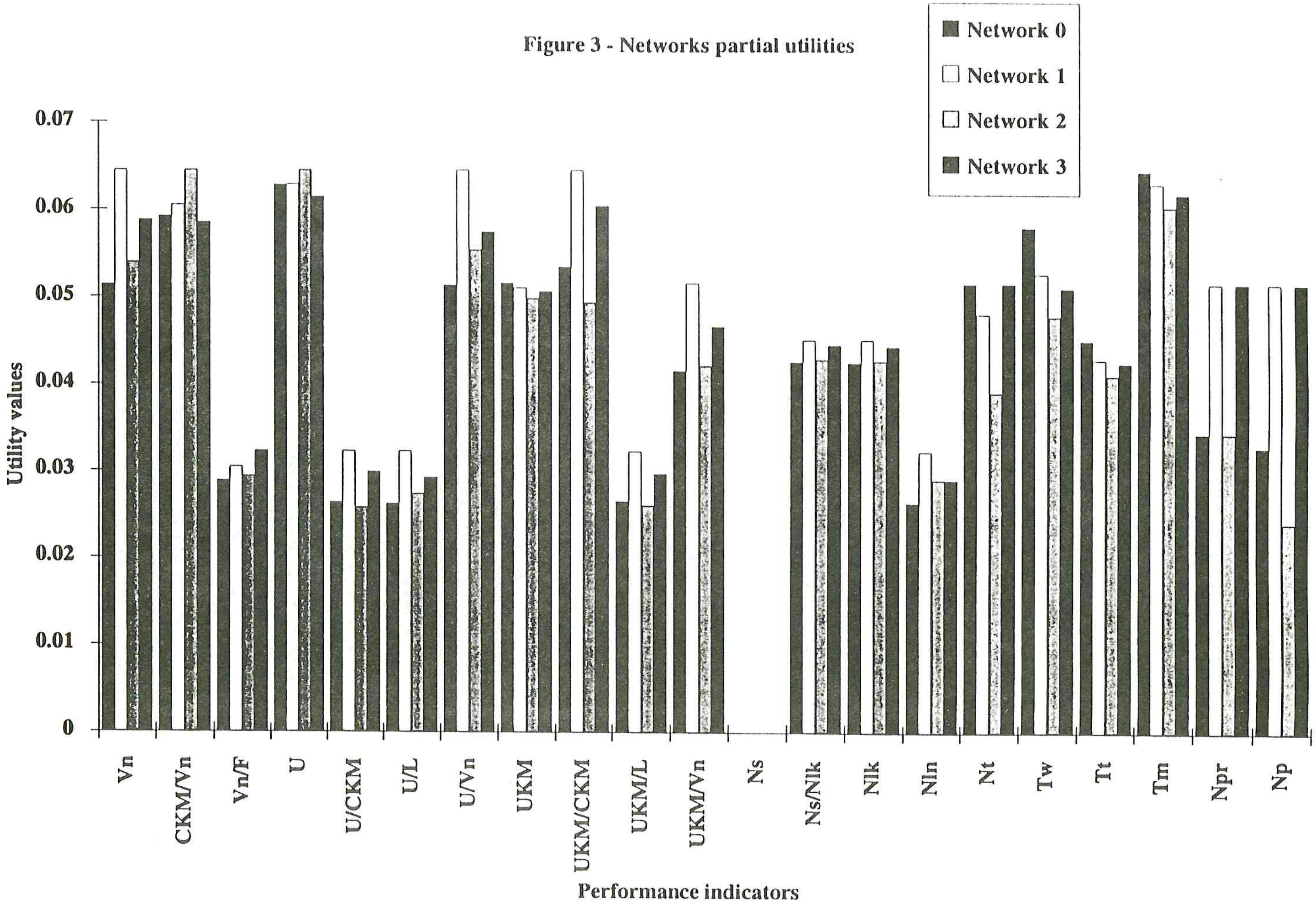


Figure 3 - Networks partial utilities



MULTI-OBJECTIVE APPROACH FOR DESIGNING TRANSIT ROUTES AND FREQUENCIES

by

YECHEZKEL ISRAELI & AVISHAI CEDER
Transportation Research Institute, Civil Engrg. Dept., Technion-Israel Institute of Technology, Haifa 32OOO, ISRAEL
Tel: 972-4-293O54
Fax: 972-4-225716

ABSTRACT

Transit route design (TRD) is considered the most complex and cumbersome problem across network route allocation problems. The wide range of the TRD's characteristics creates difficulties to formulate the problem uniquely. At the same time, the TRD's complexity of the NP-hard type creates combinatorial problems. The TRD problem is formulated as non-linear programming with mixed variables (continuous and integer). This formulation cannot be solved via known mathematical programming approaches and packages. This research provides a new and efficient approach to solve the TRD while dealing with both its complexity and its practical issues. The approach used has an impact on three components involved: the operator, the user, and the considered authority. The objectives of these three components do not always coincide. From the operator viewpoint, the system should minimize its expenses while, from the user perspective, the system should - maximize its level-of-service. This trade-off situation creates this work's optimization framework. the multi-objective programming technique that was applied has not been used, to our knowledge, for solving the TRD problem. In fact, due to the problem complexity, the ordinary mathematical programming methods cannot be used in this technique, and therefore, a new approach is provided. This new approach is heuristic in nature and divided into two phases: (a) generation of finite sets of alternative efficient non-inferior solutions; and (b) evaluation and selection of the various solutions using multi-objective preference techniques for discrete variables ("compromise programming" procedure). This paper describes a general procedure and algorithm, based on a given covering matrix, which generates "promising alternatives" for the multi-objective solution. Each alternative is also a feasible solution to that covering matrix. The matrix itself which guarantees connectivity between all origin-destination pairs of the network can be created by "column generation" procedure in former stages. This approach enables to solve the complex TRD problem. It combines mathematical programming with decision-making methods, using search and enumeration processes while performing the optimization. Thus, it is possible to encounter relatively large-scale problems (networks) which cannot be solved by other techniques.

1. INTRODUCTION

Transit route design (TRD) is considered the most complex and cumbersome problem across network route allocation problems. The wide range of the TRD's characteristics creates difficulties to formulate the problem uniquely. At the same time, the TRD's complexity of the NP-hard type creates combinatorial problems. The TRD problem is formulated as non-linear programming with mixed variables (continuous and integer). This formulation cannot be solved via known mathematical programming approaches and packages.

The importance of the improvement of an existing TRD solution is that with a relatively low investment, the whole transit operation can be significantly improved. This research provides a new and efficient approach to solve the TRD while dealing with both its complexity and its practical issues. The approach used has an impact on three components involved: the operator, the user, and the considered authority.

The objectives of these three components do not always coincide. From the operator viewpoint, the system should minimize its expenses while, from the user perspective, the system should maximize its level-of-service. This trade-off situation creates this work's optimization framework.

The optimization criteria of the operator-user-authority components consider, simultaneously, the formulated combination of: route design; timetabling (frequencies); and vehicle scheduling. The methodology used is unique in comparison to other methods, due to the consideration of daily operational elements (passenger counts, passenger load profiles, operational fleet size, etc.). This methodology enables to produce route changes and route design even for the short-range planning.

The mathematical formulation of this work is based on four objective functions -- each to be minimized. However, it is impossible to treat all functions simultaneously, and hence, multi-objective programming is being used. This multi-objective programming technique was not used, to our knowledge, for solving the TRD problem. In fact, due to the problem complexity, the ordinary mathematical programming methods cannot be used in this technique, and therefore, a new approach is provided. This new approach is heuristic in nature and divided into seven modules (see Israeli (1992)): (i) creating large set of feasible routes; (ii) creating feasible transfer paths obeying a given level-of-service criterion; the outcome of these two modules which use "column generation" procedure is a Set Covering matrix; (iii) solving the Set Covering Problem (SCP): selection of a set of routes and transfers enabling connectivity among all the network nodes with minimum length of demand paths: (iv) demand assignment while considering the various origin-destination paths. This part determines the vehicle frequencies and the timetables; (v) derivation of an estimate for the required fleet size while deciding the vehicle scheduling based on a given timetable; (vi) generation of finite sets of alternative efficient non-inferior solutions; and (vii) evaluation and selection of the various solutions using multi-objective preference techniques for discrete variables ("compromise programming" procedure).

This research deals with the last two modules and their relation to the former three modules (SCP, assignment procedure and fleet size estimation). The research provides a general procedure and algorithm, based on a given covering matrix which generates "promising alternatives" (sets) for the multi-objective solution. The relationship between the different modules is as follows: while the solution of the SCP is based on the costs which are derived from the assignment procedure, the assignment procedure itself is based on the solution of the SCP. This yields an iterative procedure, the outcome of which is the "alternative generation process."

The suggested approach enables to solve the complex TRD problem. It combines mathematical programming with decision-making methods, using search and enumeration processes while performing the optimization. Thus, it is possible to encounter relatively large-scale problems (networks) with the possibility to interact with the solution method during intermediate steps. The final results of the analysis provide transit routes with frequencies (timetabling) and the framework for vehicle scheduling.

2. PROBLEM IDENTIFICATION

2.1 Research Framework and Formulation

The transit planning process, aimed at efficient transport of origin-destination transit riders, includes four basic components performed in sequence: (a) Network Route Design; (b) Setting Timetables; (c) Scheduling Vehicles to Trips; and (d) Assignment of Drivers. In order for this process to be cost-effective and efficient, it should embody a compromise between passenger comfort and cost of service. For example, a good match between bus supply and passenger demand occurs when bus schedules are constructed so that the observed passenger demand is accommodated while the number of vehicles used is minimized.

Whereas most of the research and computer programming concern the last two components (see analysis of Ceder and Wilson (1986)), few researchers have studied the interrelationship between the scheduling components and the network design element. The interrelationship exists in two directions: (i) each set of routes yields, based on the demand, a different set of frequencies and timetables, and ultimately, the required fleet size; and (ii) the operational cost derived from the scheduling components and the passenger level of service affect the search for the optimal route design while relying on a compromise between the operator and the user.

The approach presented in the present paper considers the first three components simultaneously: route design, timetabling (frequencies), and vehicle scheduling. It combines the philosophy of the mathematical programming approaches with decision-making techniques, in order to allow the user to select from a number of alternatives.

The mathematical formulation is based on several objective functions and performance measures. First, there is a method to select the most crucial variables and parameters to effect the operation. Second, four objective functions are being while incorporating the various operational and user perspectives:

$$\begin{aligned} \min\ Z_1 &= PH \\ \min\ Z_2 &= WH \\ \min\ Z_3 &= EH \\ \min\ Z_4 &= FS \end{aligned} \tag{2.1}$$

where:

PH= Total Passenger Hours between all origin-destination pairs (defined as passengers' riding time in a bus on an hourly basis. It measures how much time is spent by passengers on buses between the two nodes);

WH= Total Waiting Time between all origin-destination pairs (defined as the amount of time on a bus on an hourly basis. It measures how much time is spent by passengers on buses between the two nodes);

EH= Total Empty Space Hours on all routes of the network (defined as the unused seats in a bus on an hourly basis. Empty Space Hours measure to what capacity buses are used);

FS= Fleet Size (number of buses needed to provide all trips along the chosen set of routes).

The complete formulation, including the constraints of the network design problem, can be found in Israeli (1992); Ceder and Israeli (1992). The nature of the overall formulation is non-linear (non-linear and mixed integer programming). Its analog problem is the generalized network design problem described by Magnanti and Wong (1984), with an NP-hard computational complexity. Thus, conventional approaches are incapable of providing a solution even with a relatively high degree of simplification.

2.2 Characteristics and Application of the Multi-Objective Programming

Out of general mathematical formulation, a need appears for the minimizing of the four objective functions : Z_i, i=1,...,4. In fact, due to the conflict between the objective functions, it is impossible to arrive at an ideal solution incorporating a simultaneous minimization of all Z_i, i=1, ...,4. This conflict exists in all problems of transferring demands in networks in both the level of transportation planning (Current et al., (1987)) and in public transport planning (Janarthanan and Schneider (1986)). The fault lies in the undefinable nature of a reasonable objective function, including all factors of relevance which can be commonly accepted (Steenbrink (1974)). Since the four objective functions Z_1 up to Z_4 are complete and expressed in different units, it is complicated and perhaps undesirable to combine them into one unit scale without an accompanying information loss. Thus, to guarantee their separateness, we need to treat the problem as one in a multi-criteria mode. This problem entails a trade-off between the four objective functions. Thus, in the two-dimensional plane, the devaluation of one will add to the value of the other. Alternatively, a simultaneous improvement is a possible provision, bringing up the cost of a whole network system (a change in the basic condition). Hence, there can exist no optimal solution, but a variety of a compromise solution between the objective functions.

The choice of the "best" compromise can only be facilitated by the establishment of various solutions, themselves created by due process. Fricker and Shanteau (1986) stress that, beyond the mathematical selection process, there needs to exist a various options set-up, not a single optimal option, to be of genuine assistance to the decision-makers in their comprehension and subsequent selection. Such decision-making tools which apply multi-objective programming in general networks are described by Ignizio (1993).

In a transportation system, as there seem to be few models of multi-objective planning. Current and Min (1986), while surveying such studies, noted a recent increasing usage of the multi-objective approach in planning transport problems. They determine this to the development of heuristic techniques for networks and computer hardware advancement facilitating the ability to address all angles of the multi-aspect problem. Earlier techniques (dating from the mid-seventies) tended to consider evaluating alternatives instead of creating them.

From a review of relevant studies (see Israeli (1992)), we realize that none combine a multi-objective function approach which successfully addresses all aspects of the problems of public transport route planning in its processes to the total definition of the question.

Three studies describe a process of creating alternatives lacking evaluation and the usage of any multi-objective programming models. Fricker and Shanteau (1986) studied operation strategies for a small town with a given set of routes. The process, based on producing about 2O alternatives, was aided by an "intuitive" algorithm. Pogun and Satir

(1986) checked operation strategies for exclusive public transport lanes with manually predetermined alternatives which were evaluated using a simulation model. Tadi et al., (198O) used an economical costing model to develop operational alternatives for a public transportation network. All three studies employed manual analysis of the differing alternatives from the point of view of the user/operator. However, they excluded a selection for the "best" compromise.

Other studies used multi-objective programming techniques. Some merely created alternatives while others evaluated their results. Tzeng and Shiau (1988) described an analysis for determining different programming parameters, while keeping the solution process based on formulating two arched objective functions and deriving them. Alternatives are presented accompanied by performance measures. However, there is no evaluation process, and consequently no decision-making. Current et al., (1987) described location of a single route with the employment of the weighting approach. This provides alternatives yet lacks an evaluation process, thus leaving the decision to the decision-maker. Lee and Moore (1977) described a model for school bus routes using the Goal Programming approach. This included the evaluation of the resulting alternatives. Janarthanan and Schneider (1986) showed the evaluation of given designed alternatives for public transport networks, using the concordance analysis approach. In their method, there was no possibility of generating new alternatives outside those given.

Two studies dealt with air routes. Flynn and Ratick (1988) described the coverage problem for areas requiring air service. The basic decision variables in this instance were connecting or disconnecting population centres to service centres (main airports). The connection was represented by a one arc secondary route. The method utilized here was the Constraint Method. The evaluation and choice of the solution were achieved quantitately with the exception of mathematical tools.

Teodorovic and Krcmar-Nozic (1989) calculated frequencies for given air routes. The uniqueness of this study is in its treatment of more than two objective functions (namely, three), in the process of creating alternatives. The research was based on a heuristic method to produce random creation of possible solutions. The most appropriate choice was arrived at by adjusting the preference to the mini-max problem between the various alternatives.

The multi-objective programming problem can be classified into two differing alternative characteristic types: discrete problems and continuous problems. The discrete type problems are based on a number of alternatives from which one is preferred. However, the continuous mode problems require a model entailing decision variables, constraints and objective functions for creating suggested alternatives. Such variables may have any value from a given successive value structure (Cohon (1978)).

The techniques employed can change depending upon the timing and the amount of information preference made available by the decision-maker. For example, if beforehand, the preference relation among the objective functions is expressed, the problem is reduced to that of a one objective function. Accordingly, through the relative alternatives, the objective function can be weighted to a single function. This condition is normally unrealistic. Conversely, if no preference information is prerequested, the planner may attempt to cover all useful alternatives within the extreme parameters of the multi-objective problem, presenting a comprehensive set of alternatives to the decision-maker. However, all these possibilities can prove too numerous to produce classification strata necessary to effect an informed choice.

Between these two extremes there exist the partially creating techniques providing subsets of useful solutions for the decision-maker. Within a reasonable time span, they will have more information concerning the problem, along with its possible solutions. The principal of optimum choice in such a case is replaced by one of utilization.

The two stages in resolving the multi-objective programming problem are:
1) building the non-inferior solutions (the utilization curve); 2) choosing solutions from within that curve; in short, recognizing the "compromised set." The non-inferior solutions'set -- NIS -- expresses the collection of points in the objective functions' space in such a way that any improvement in one single objective function could only be achieved with a simultaneous damaging or reduction of, at least, one of the other objective functions. Thus, x^* will belong to NIS if another solution will not yield $x \varepsilon X$, so that :

$$Z_i(x) \leq Z_i(x^*) \qquad , i = 1,2,\ldots,p$$

with at least one inequality, while $Z_i(x)$ marks the value of the objective function in the point of x. The set of points belonging to the NIS is a subset of the feasible solution set x. An example of the two-dimensional case is demonstrated in Fig. 1. The extreme points which bound the NIS are the solutions of one-dimensional optimization problem of $Z_i(x)$ and $Z_j(x)$.

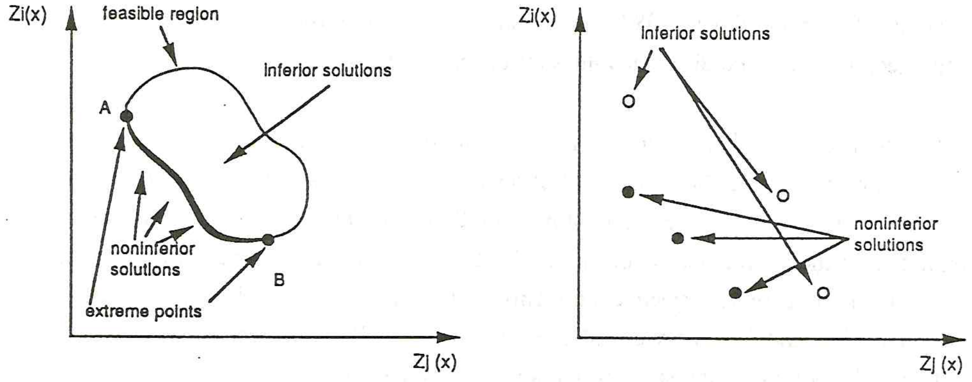


Fig. 1: A scheme of non-inferior and inferior solutions: (1) continuous; (2) discrete problems

The solution techniques of multi-objective programming are based on the fact that in most practical problems, the dimension in the space of the objective functions is much smaller than those of the decision variables (p<n); hence, it is easier to perform the objective functions' space while only occasionally referring to that of the decision variables.

The choice of multi-objective technique demanded appraising the two stages: creating efficient solutions and choosing the compromise solutions whilst evaluating them. The problem when analyzed is of non-linear nature (concave), containing integer variables. These include the complex form of NP-hard. Such characteristics prevent the use of mathematical programming techniques inherent in small dimensional problems. It is therefore impossible to use known multi-objective programming techniques which deal with problems of a continuous nature (see the analysis of Teodorovic and Krcmar-Nozic (1989)). The methods employed for these types of problems are based on a full mathematical programming; both for finding the utilization curve (NIS) and for the process of choosing the compromise - solution. Most of the existing models according to the available techniques tend to fit linear problems. Only recently has attention been paid to the resolution of non-linear problems (see Current and Min (1986); Fandel and Spronk (1985); and Hwang and Masud (1979)). The treatment of multi-objective problems containing integers started later. Presently, only a few techniques exist (see Lazimy (1985), Rasmussen (1986), and Gabbani & Magazine (1986)).

Consequently, Tzeng and Shiau (1988) in their study used a simplified mode within which a simulating technique derived different values for the binaric variables.

Proceeding from this problem, it is impossible to construct a set of continuous NIS, thus a heuristic method for the creation of efficient solutions must be maintained. In this method, an absolute number of alternative solutions is created. These will be an approximation to the continuous set of the NIS. Its evaluation will be made with reference techniques to discrete multi-objective problems. This method fits both Janarthanan and Schneider's (1986) and Teodorovic and Krcmar-Nozic's (1989) research. These studies define the problem of public transport evaluation as a discrete one.

The latter study converted the mathematical programming of the continuous problem into a heuristic approach which randomly select points from a large set of feasible points whilst using the Monte Carlo method to solve the IP. The process repeats itself with different random small samples from which the "best" solution is selected as an estimator to the global optimum. Analyzing their approach shows two main drawbacks: (1) there is no certainty to receive the compromise set; (2)' the solution depends on a probability function which itself is unknown and depends on various assumptions. Relating to these conclusions, it was decided in our research to use a different heuristic process in order to generate a large set of efficiency points which can estimate the continuous utility curve NIS. Examining the different multi-objective techniques yields the use of the "compromise programming" technique.

3. RESEARCH APPROACH

3.1 Column Generation Approach

As described in section 1, a heuristic modular method is provided in order to overcome the mathematical complexity of the problem. The entire process is based on the first two modules which use a "column generation" approach to define the covering matrix of all possible travel paths in the network.

The process is based on a mapped network of routes and transfer paths. This incorporates within the network guaranteed connectivity under the route length criteria. It includes the deviation percentage allowed, using the projected minimum travel time (private cars) and a maximal of expected bus route transfers. This network is described comprehensively in Israeli (1992). However, the large set of routes and transfers is likely to contain many overlapping segments (of routes and transfers). An overlapped segment is one that serves O-D pairs that are wholly served by other routes and/or transfers. An overlapped route comprises segments which are all totally overlapping. The latter is treated as follows:

The system creates minimal set(s) of routes and their related transfers, such that connectivity between nodes is maintained and their total deviation from the shortest path is minimized. The problem is defined as a Set Covering Problem (SCP) which is hard to solve (see Minieka (1978); Syslo et al., (1983)). The SCP can determine the minimal set of routes from the matrix of the feasible paths (routes and transfers). In this matrix, which is shown in Table 1, each row represents either a feasible route or a transfer. The "1" in the matrix is inserted whenever an O-D demand can be transported by the route or transfer, and "O" otherwise. (Terms will be defined hereafter).

Table 1: Matrix A Configuration of Feasible Paths

	$r \in R$	$tr \in TR$			
		$k_{tr} = 1$	$k_{tr} = 2$		$k_{tr} = n$
$i,j \in N$	$\{a_{ij}^{r}\}$	$\{a_{ij}^{tr}\}$	$\{a_{ij}^{tr}\}$		$\{a_{ij}^{tr}\}$
	$\{C_r\}$	$\{C_{tr}\}$	$\{C_{tr}\}$		$\{C_{tr}\}$

The word "covering" refers here to at least one column with "1" in each row. The transfers are combined columns in the SCP matrix and therefore increase the complexity of the problem. No solution appears for this problem in the literature. The mathematical formulation of the SCP-transfer problem results in a non-linear programming with integer variables. Thus, if we let c_r be the cost of a direct route and c_{tr} the cost of a transfer (each one is referred to in a single column), then the formulation is:

$$\min \left[\sum_{r \in R} c_r x_r + \sum_{tr \in TR} c_{tr} \underset{r:tr}{\pi} x_r \right] \tag{3.1}$$

s.t.

$$\sum_{r \in R} a^{r}_{ij} x_r + \sum_{tr \in TR} a^{tr}_{ij} \underset{r:tr}{\pi} x_r \geq 1 \qquad \forall i, j \in N \tag{3.2}$$

$$x_r = \begin{cases} 1 & \text{route } r \in R \text{ in the solution} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$$\text{when} \quad a^{tr}_{ij} = \{0,1\} \quad \forall tr \in TR, \qquad a^{r}_{ij} = \{0,1\} \quad \forall r \in R$$

Note that in the covering matrix A, a link exists for origin-destination pair via a direct route ($r \varepsilon R$) or transfer ($Tr \varepsilon TR$), and:

$$\sum a^{r}_{ij} + \sum a^{tr}_{ij} \geq 1 \tag{3.4}$$

The notation r:tr signifies that the route r consistently shows in the transfer path tr. k_{tr} denotes the number of bus changing in a transfer path. The costs c_r and c_{tr} at this stage, before assigning the demand, describe the lengths of the travel paths: direct routes and via transfers. Hence, the solution of the covering problem will yield the travel path network (direct and indirect), with the minimum travel times.

An algorithm developed for the above formulation (which appears in Israeli (1992) and to be published elsewhere), has been tested with a random network. The results showed its high efficiency under the criteria of accuracy and running time.

3.2 The Entire Iterative Process

At this stage, the real costs of matrix A's columns (routes and transfers) are not known. These costs are similar to the optimization criteria which were defined in (2.1), but associated with each column. The costs can result from the demand assignment procedure which, by itself, is based on the minimal set of routes, i.e., the SCP solution. That is why it is impossible to solve the SCP to minimum real costs, but to minimum travel time.

Thus, at this stage, commences the iterative process of constructing efficient alternatives for resolving the multi-objective problem. This comprises three main stages:

Stage 1: Producing a minimal set of direct/indirect travel paths (routes/transfers) affirming the connectivity of the network. The problem to be solved is known as SCP (Set Covering Problem).

Stage 2: The execution of the assignment process over the covering set which was created in the former stage. The outcome of the stage is: bus frequencies of the set of routes, passenger load profiles, demand assignment across the set of routes, and the optimization parameters PH, WH, EH. At the same time, the minimim fleet size required to meet the demand as well as to satisfy the determined frequency on each route is estimated. The method used for evaluating the fleet size is based on the Deficit Function theory proposed by Ceder and Stern (1981). The final outcome of this stage is the value FS which represents the operational component's minimum required fleet size.

Stage 3: A deletion of the detrimental variables (routes/transfers) to the value of the solution acceptance of a new (diminished) set of travel paths (routes and transfers) within the network, which results in a diminished matrix A, and back to stage 1. The process executed in stage 3 guarantees two main issues: (1) there will always be a cover to the diminished matrix A; (2) previous alternatives will not be repeated.

Once the criteria reach the desired number of alternatives, the alternative creation process results in the multi-objective problem being resolved, up to the minimizing of Z_1 up to Z_4.

This paper deals mainly with the relationship between stages 1 and 3, and in resolving the preceding multi-objective problem. The second stage (demand assignment) is not detailed herein; for more information see Israeli (1992).

3.3 Alternative Generation Process

The described process generates a set of feasible and efficient points in the field of the four objective functions. Such a point is typical for a feasible alternative solution which represents a set of routes. These points are an approximation of the efficient points, yet unknown, which result from solving the continuous problem. This method of generation results from the approach of "compromised programming" chosen for evaluating the results and choosing from them (see 3.4). The criteria at each stage were to reach the minimal distance (metric value) from the present ideal solution (from the minimal theoretical point of the four objective functions at the same time).

The process separates into two problems: master problem and sub-problem.

The master problem deals with incremental progress from one solution to the other in the generation process. The problem deals with building the covering matrix A of the possible routes and transfers, and solving it in such manner that different solutions will be created (new sets of routes transferring the network demands). The initial point is the covering sets which results from the first covering problem (described in (3.2)), for which the values of the objective functions are calculated according to the assignment process and the estimation of the operating vehicle fleet.

At this stage, the ideal solution is defined anew, as well as the set with the minimal distance to the ideal point. The variable most detrimental to the set's metric value cost is deleted from this set. Recovering matrix A is effected, and the process repeats itself. The minimal distance is dynamic, since the ideal point can be changed during the iterations.

The sub-problem termed SDP (Set Deletion Problem) deals with producing feasible solutions for the covering problem SCP, of the master problem. The question deals with two elements: 1) If a variable column is deleted in the master problem (or a combination of columns) from the covering matrix A, there could arise a situation where A will have no covering. In such a case, a column should be returned (or a partial combination of columns), to the matrix; 2) In the covering process for the master problem, it is essential not to enter an endless loop; this means to receive already existing sets (solutions). The SDP problem therefore, is dealing with a minimal deletion of columns from matrix A in a certain iteration so that by solving the SCP problem within the master problem (in a diminished A matrix), solutions will not be reiterated, and there will always exist coverage for A. The SDP problem is formulated as a set covering problem (non-linear), and proper resolution techniques are suggested. For clarification, it is emphasized that a few equivalent terms have been used, and have the same meaning: solution alternatives; sets (of routes); feasible solutions; feasible points (in the four objective functions' space).

3.3.1 The master problem

The algorithm described examines in each stage the sets of feasible points $\{(Z_1, Z_2, Z_3, Z_4)\}$ whilst generating a new solution which can guarantee approaching the ideal point. If such solutions exist, the outcome will comprise all sets of efficient solutions (efficient points in the four dimensional space).

define:

CAN = a group of all sets of routes (solutions of SCP) which are examined and are candidates for changing in the alternative generation process;

SOL = a group of all sets of routes which were selected to solve the multi-objective programming;

n_c = number of sets in CAN;

n_s = number of sets in SOL;

n_{max} = an upper bound for the required number of sets;

n_{set} = an upper lexicographic bound for the selection of a column out of a re-examined set;

L_s^k = metric distance from the power of s of the set $k \varepsilon CAN$ (see definition of section 3.4).

Algorithm's steps:

Step 0: Selection of criteria:
Assign initial values to n_{max}, n_{set} and s.
Step 1: A group of candidate sets:
Solve the SCP of the initial matrix A. On the solution's set(s), execute the demand assignment process and calculate the estimator of the minimum fleet size. A feasible point(s) (Z_1, Z_2, Z_3, Z_4) was created. Group the sets in CAN and in SOL (at this stage $n_s = n_c$).
Step 2: Initial conditions of the algorithm:
If $n_s > n_{max}$ take SOL as the final feasible solutions. Otherwise, go to step 3.

Step 3: Finding the ideal point in the present iteration:

The sets in CAN yielded:

$$Z_i^* = \min \quad Z_i \qquad ,i = 1,\ldots,4$$

and define point $\left(Z_1^*, Z_2^*, Z_3^*, Z_4^*\right)$

Step 4: Metric values:

For all sets in CAN, calculate the metric values L_S from the ideal point (see formulas in section 9.3).

Step 5: Choosing the candidate set:

From the sets in CAN choose k' which keeps a minimal "potential metric distance" from the ideal solution, which means:

$$L_s^{k'} + \Delta L_s^{k'} = \min_{k \in CAN} \left(L_s^k + \Delta L_s^k\right) \qquad (3.5)$$

ΔL_s^k shows the metric distance difference $L_s^j - L_s^k$ when the set j has been created last as a result of candidate set k in former stages. (The set which has not been generated will be marked $\Delta L_s^k = 0$). This sign marks the production potential of metric distances chosen by set k, and its aim is to prevent the degeneration of the problem. Without this sign, the candidate set would be the one showing:

$$\min_{k \in CAN} L_s^k$$

and then there is a possibility of obtaining "bad" sets from such sets {j} that have $L_s^j > L_s^k$ which will result in re-selecting set k in all the iterations, thus preventing the possibility of improving the solution. The correction in the formula enables the selection of another set with higher value of L_s^k, but with potential to generate sets with relatively low metric values.

Step 6: Candidate set's status:

Check if the set selected in step 5 was executed in former iterations (i.e., was a candidate from which other sets were generated). If so, go to step 8; otherwise continue.

Step 7: A candidate set executed for the first time:

Select the column of a singular route about to be deleted from the covering matrix: Select from the set the column of a singular route r' which delineates:

$$\hat{L}_s^{r'} = \max_r \quad L_s^r \tag{3.6}$$

where $\hat{L}_s^{r'}$ is the metric value of the route (after the assignment process) from the set's ideal point for a covering unit, meaning the metric value divided by the number of pairs "i,j" (rows in the covering matrix), those served by this route and the resultant transfers.
The set's idea point is:

$(\min\ PH_r, \min\ WH_r, \min\ PH_r, \min\ FS_r)$

and the index r related to the same optimization criteria previously defined, but which results from the load profile of route r only. FS_r is calculated by the portion usage of route r from the total FS.
Step 8: A candidate set which is re-executed:
Select in the candidate set the preferred column to be deleted according to the hierarchial order of decreasing metric values (see Eq. (3.6)). There are two possible cases: (1) A column of a single route which was not performed in former stages (was not chosen as a candidate for deletion); (2) A combination of columns which was not performed in former stages. The hierarchial order is based on a binary number of n_{set} digits in which "1" denotes the column to be selected and "0" otherwise.
Step 9: A candidate set which cannot be re-executed:
If one of two conditions is obtained:

(1) The priority of columns to be rejected is given by a n_{set} digit binary number with "1" in all its digits. In such a case, the n_{set} bound was completely extracted;

(2) The priority of columns to be rejected is given by a binary number with less than n_{set} digits, of which the right digit is "0" and the others are "1." The set itself compounds of n_{set} columns. In such a case, the set is totally extracted, and if the lexicographic deletion process will be continued, the set will be aborted in the next stage.

Step 10: Deletion of columns from covering matrix A:
Present the column(s) of the route(s) selected in steps 7 or 8 as columns in SDP matrix (see section 3.3.2).

Step 11: Examining SDP solution:

If SDP performs a feasible solution, go to step 12.

Otherwise, if the candidate set is still in CAN, replace the columns which were deleted and go to step 8. Otherwise -- go to step 15.

Step 12: Solving SCP in diminished matrix A:

Delete from matrix A the columns of routes r which are the solution of SDP and the columns of transfers tr associated with them. Solve SCP to diminished matrix A. A definite solution will be obtained.

Step 13: Obtaining another alternative:

In the set obtained in step 12 (the solution of SCP), the demand assignment procedure will be executed, and hence the calculation of the minimum fleet size. These yield a new point (Z_1, Z_2, Z_3, Z_4) in the four-dimensional space. Add the new set to CAN and to SOL.

Step 14: Process terminating -- first possibility:

$n_s = n_s + 1$; if $n_s < n_{max}$ go to step 3; otherwise the process is terminated. n_{max} feasible alternatives were achieved.

Step 15: Process terminating -- second possibility:

If $CAN = \phi$ go to step 5; otherwise the process is terminated. $n_s < n_{max}$ feasible alternatives were achieved.

It will be noted that in each iteration of the described algorithm, SOL will certainly increase while CAN might (not necessarily) decrease or even receive the value $CAN = \phi$, (means $n_s \geq n_c$). n_c decrease is subject to steps 5 and 11. The number of alternatives to be generated depends on the density of matrix A and the bounds selected in step 0.

3.3.2 The sub-problem: SDP

The SDP checks the columns about to be deleted from matrix A in the master problem. A different matrix is defined whose columns are those deleted from sets in former iterations and from the present candidate, and its rows are the fitting sets. The dimensions of the matrix increase as the iteration progresses within the master problem, so that in each iteration a row must be added for marking the examined set, and a column may be added or replaced. Thus, the covering problem is solved, resulting in a minimal amount of columns to be rejected from A, so that there will be no repetition sets (which exist in SOL). The SDP problem has definite covering, and according to the chosen columns, two kinds of solutions emerge: 1) A feasible solution. -- The chosen columns are deleted from matrix A, and A has coverage; 2) An infeasible solution. -- The chosen columns will not permit a covering for A. The second case is derived from a possible combination of columns in SDP that exclusively cover one entire row in A (meaning the values of 1 are derived only for these columns, on at least one row, while for the rest of the columns, the values are 0). Such a combination, designated "A Unique Combination," if entirely rejected from A will result in 0 values at least on one row, and as a result will avoid coverages. The formulation and solution of SDP attempt to prevent the entrance of the "unique combination" to the solution. Lacking a choice, it would be possible to perform in the algorithm of the master problem, a backtrack for transferring the rejected route columns, as described in step 11 of the master problem.

Define $E = \{\{e_{ij}\}, \{e_{if}\}\}$ the matrix to solve the SDP (its binary parameters will be explained later). The stages to solve the problem are as follows:

Stage 1: Matching matrix A with the potential matrices E. Scan the rows of matrix A and mark to each row the columns which cover this row exclusively. Such columns can be presented by single routes or transfers, $R' \subset R$ and $TR' \subset TR$ accordingly. That condition is given by:

$$\sum_{r \in R'} a_{ij}^{r} + \sum_{tr \in TR'} a_{ij}^{tr} = \sum_{r \in R} a_{ij}^{r} + \sum_{tr \in TR} a_{ij}^{tr} \geq 1 \qquad \forall i, j \in N \qquad (3.7)$$

Derive the "unique combinations" to matrix A. These combinations will comprise only single route columns, both those directly included in (3.7) and constructing the transfer paths included in (3.7). A collection of "unique combinations" -- UCOM was obtained.

Stage 2: Constructing matrix E in a certain iteration of the master problem.

sub-stage 2.1: Column generation.

(a) Single route columns (J):

These columns are the route columns of matrix A which fulfil condition (3.6) of each set in SOL and the candidate set (step 7). These are the columns most detrimental to the ideal points of the different sets. If, in a certain iteration of the master problem, step 8 takes place instead of step 7, columns in matrix E will be replaced. The candidate column(s) to be selected in step 8 will be added to matrix E. Columns already existing in matrix E and belonging to the set chosen in step 8 which did not fulfil the condition (3.6) for other sets in SOL, will be deleted. If step 8 selects a combination of columns, they will be unified to one column with value 1 to the binary parameter (see (3.8)), if at least one column which comprises this combination has 1 value in this parameter.

(b) "Unique Combinations" columns:
Check each "unique combination" which belongs to UCOM if it is included in column collection J. If so, add this combination as a column to matrix E (in addition to the single columns of the routes of this combination which are included in J).

sub-stage 2.2: Row generation
The rows are all the sets in SOL and the candidate set (which was selected in steps 7 or 8). The SDP solution will prevent these sets from appearing in the solution of the master problem. Each set (row) in the matrix is represented by at least one route (column). The binary parameter is:

1 if route $i \varepsilon I$ exists in set $j \varepsilon J$

$$e_{ij} = \begin{cases} 1 & \text{if route } i \in I \text{ exists in set } j \in J \\ 0 & \text{otherwise} \end{cases} \qquad (3.8)$$

$$e_{if} = 0 \quad , i \varepsilon I , f \varepsilon F \qquad (3.9)$$

The "unique combinations" column parameter is zero to each row of the matrix. This is because the influence of such columns is not on the covering, but on the objective function value explained herein.

sub-stage 2.3: Column costs
The column collection J will be divided into three groups:
$J = \{J_o \cup J_1 \cup J_s\}$ while:

$J_o =$ route columns which do not belong to the selected set in the present iteration;
$J_1 =$ route columns which belong to the selected set in the present iteration;
$J_s =$ the column in the selected set chosen as candidate for deletion (steps 7 or 8).
The cost definition will be accordingly:

$$\{c_j\} = \{c_j^o\} \cup \{c_j^1\} \cup \{c_j^s\} \qquad , j \in J$$

(a) A column of the selected set chosen for deletion:

$$c_j^s = -\infty \qquad ,j = J_s \tag{3.1O}$$

Since this column must be deleted from matrix A, solving SDP to minimum will certainly contain this column in the solution.

(b) Columns not belonging to the selected set:

$$c_j^0 = 1 \qquad ,j\varepsilon J_o \tag{3.11}$$

(c) Columns belonging to selected set and also existing in the other sets of the matrix (excluding the column described in (a)):

Two possibilities exist:

1. There is no preference policy in the SDP solution for the columns included in the selected set over the other columns. Thus:

$$c_j^1 = 1 \qquad ,j\varepsilon J_1 \tag{3.12}$$

2. There is a preference policy for the columns of the selected set. In this case, there is a need that matrix A should contain as many columns as possible which belong to the selected set. While solving the SCP of matrix A, it increases the chance of obtaining a set similar to the one selected (a new point in the vicinity of the efficient point selected in the Z_1, Z_2, Z_3, Z_4 space), thus, in order to prevent, if possible, the columns of the selected set appearing in the SDP solution. The columns which are included in J_1 are put in order of decreased metric values from the set's ideal point (see step 7 of the master problem), while the hierarchial order is $m = 1,2,...$. Hence, column m obtains

$$c_j^1 = 2^{m-1}\left(\sum_{j\in J_0} c_j^0 + 1\right) \qquad \forall j \in J_1 \tag{3.13}$$

By this formula, each column in J_1 will have a higher cost compared to a column in J_0. Also, for different columns in J_1, the lower the metric value, the higher is the cost in matrix E. This approach enables the solution of SDP to minimum, to contain, if possible, columns with low costs. This means columns in J_o instead of J_1, and in J_1 columns with higher metric values.

(d) "Unique Combinations" columns

Their costs c_f are defined by:

$$c_f = +\infty \quad , \; f \varepsilon F \tag{3.14}$$

The high value of the costs will prevent, if possible, the "unique combinations" columns from appearing in the SDP solution (if an alternative covering by other column exists). Matrix E configuration is described in Table 2.

Table 2: Matrix E Configuration

	$j \in J$			$f \in F$
	$j_o \in J_o$	$j_1 \in J_1$	$j_s \in J_s$	
$i \in I$	$\{e_{ij})$	$\{e_{ij}\}$	$\{e_{ij})$	$\{e_{if}\}$
	$\{c_j^o\}$	$\{c_j^1\}$	c_j^s	$\{c_f\}$
	c_j			

Stage 3: Mathematical formulation for SDP.

The set-covering problem of matrix E is formulated as non-linear programming with integer variables:

$$\min \left[\sum_{j \in J} c_j x_j + \sum_{f \in F} c_f \prod_{j:f} x_j \right] \tag{3.15}$$

s.t.

$$\sum_{j \in J} e_{ij} x_j \geq 1 \qquad \forall i \in I \tag{3.16}$$

$$x_j = \begin{cases} 1 & \text{if column } j \in J \text{ is in the solution} \\ 0 & \text{otherwise} \end{cases} \qquad (3.17)$$

and exists

$$\sum_{j \in J} e_{ij} + \sum_{f \in F} e_{if} = \sum_{j \in J} e_{ij} \geq 1 \qquad \forall i \in I \qquad (3.18)$$

The marking j:f equivalent to $j \mid j \cap f \neq \phi$ means that column $j \varepsilon J$ takes part in the "unique combination" $f \varepsilon F$.

Stage 4: Solution techniques to solve the SDP.
The formulation (3.15) - (3.18) is equivalent to the non-linear SCP formulation of matrix A (3.1) - (3.4). Thus, solution techniques applicable to solve the covering of A can also be useful to E. It must be emphasized that the small dimensions of matrix E (in size order compared to A) enables to solve the SDP with relatively simple techniques.
The possible solution techniques:

(a) Mathematical approach

The NLP can be solved via relaxation of the integer variables ($o \leq x_j \leq 1, j \varepsilon J$). The small dimensions of the matrix make it possible to round up the non-integer solutions manually, or apply common techniques. It should be emphasized that if matrix E does not contain "unique combinations" columns, the problem is formulated as the usual IP which can be solved by integer programming and by all the "classic" techniques to work out the SCP (see Minieka (1978); Syslo et al., (1983)).

(b) Trial and error approach

This approach is applicable to the non-linear problem which contains the columns F. The NLP problem with its integer variables is converted into a series of (linear) IP problems, and then an experiment to discover a solution not containing a "unique combination" column is performed. The approach is divided into two steps explained herein. While step b.1 is not essential, but can be utilized as a "gamble" for finding the solution (then the long step b.2 will be unncessary), the process can always be commenced in step b.2 without a previous step:

b.1: solving SCP to sub-matrix J:

Set in the formulation (3.15) - (3.18) $c_f = o$, $f \varepsilon F$ (i.e., $F = \phi$). The formulation is converted into an IP problem with only single route columns J. Solve the problem by common IP techniques and examine the columns in the solution. If these columns $j^* \varepsilon J$ do not compound any of the $f \varepsilon F$ columns in matrix E, then a feasible solution (without "unique combinations") was achieved to SDP. Otherwise, continue to step b.2.

b.2: solving n IP problems due to partial elimination of "unique combinations" columns

Set in the formulation (3.15) - (3.18) $c_f=o$, $f \varepsilon F$. The formulation is converted into an IP problem for sub-matrix J. Construct n IP problems in the following manner:

(1) Define the group of columns j which included in each "unique combination" column f in matrix E. The total are $k = |F|$ combinations (equal to the number of columns f) and in each n_k routes which belong to j.

(2) In each "unique combination," choose one column of route j, and define its cost as: $c_j = + \infty$. The total number of combinations is:
$n \leq \pi_k n_k$
while each time a new route column is selected from each "unique combination" with infinite costs. Since it is possible that some of the columns j belong simultaneously to a number of "unique combinations," then the number of the actual combinations can be less than π n_k which are the upper bound. The selection of such k combinations is performed by the solution of a sub-problem named UCP (Unique Combination Problem) to be explained in the continuance. The infinite value of the costs prevents the corresponding columns, one of each "unique combination," from taking part in the solution if the matrix is "rich" with columns $j \varepsilon J$ with costs $c_j << \infty$ that can be included in the covering solution. The costs of the other columns remain unchanged, as described in stage 2 of SDP.

(3) Solve the n problems with common techniques of IP or SCP. Two cases might occur:

(i) All n problems achieve a non-feasible solution. This means that columns j^* in the solution compound completely at least one of columns f. In such a case, the solution contains at least one "unique combination."

(ii) At least one problem out of the n achieves a feasible solution (the solution does not contain a "unique combination."). From all these feasible solutions, select the one with the minimal value to the objective function (3.15) (in which $c_f = o$, $f \varepsilon F$ still exist).

A sub-sub problem: Unique Combination Problem - UCP

The problem deals with finding all the combinations of routes which are examined for deletion purposes from matrix E, in order to avoid the "unique combinations" of being in SDP solution for sub-matrix J.

Define a new matrix $U = \{u_{ij}\}$

The rows = a list of all columns $f \varepsilon F$ of matrix E;

The columns = a list of all routes $j \varepsilon J$ which are included in all "unique combinations" in matrix E (yields $j \cap F \neq \phi$)

and the binary parameter is defined by:

$$u_{ij} = \begin{cases} 1 & \text{if a route column } j \text{ belongs to combination } i \\ 0 & \text{otherwise} \end{cases} \qquad (3.19)$$

The solution finds all the covering combinations of matrix U (using enumeration). The number of possible covering of U is the number of times, n, which SDP will be solved by trial and error. Those routes which take part in the covering solution of UCP in a certain iteration receive the value $+\infty$ for their costs, while solving the corresponding IP problem (one of the n problems) of matrix E as described earlier.

(c) Heuristic Approach

Since the SDP formulation of matrix E is similar to the non-linear SCP formulation of matrix A, it is possible to use the heuristic approach (algorithm), which was developed, to solve the non-linear SCP problem (Israeli (1992)). The "unique combinations'" columns in E are equivalent to the transfer columns in A, and since they all receive $e_{if} = o$, they will not influence the covering, but the costs.

(d) Complete Enumeration

By relaying on the small dimensions of matrix E (especially in the first iterations of the master problem), it is possible, in the significant number of cases, to find its covering by a regular exchange of columns and examining the created covering. In the collection of all possible covering, the one with the minimum costs will be selected. A finite value for the costs signifies that the problem has a feasible solution, while infinite value denotes a non-feasible solution (there are columns $f \varepsilon F$ in the solution). It will be emphasized that a complete enumeration procedure can be applied also as a solution technique for the trial and error approach (section (b)).

Stage 5: The linkage to the master problem.
The solution of SDP yields one of two possibilities:

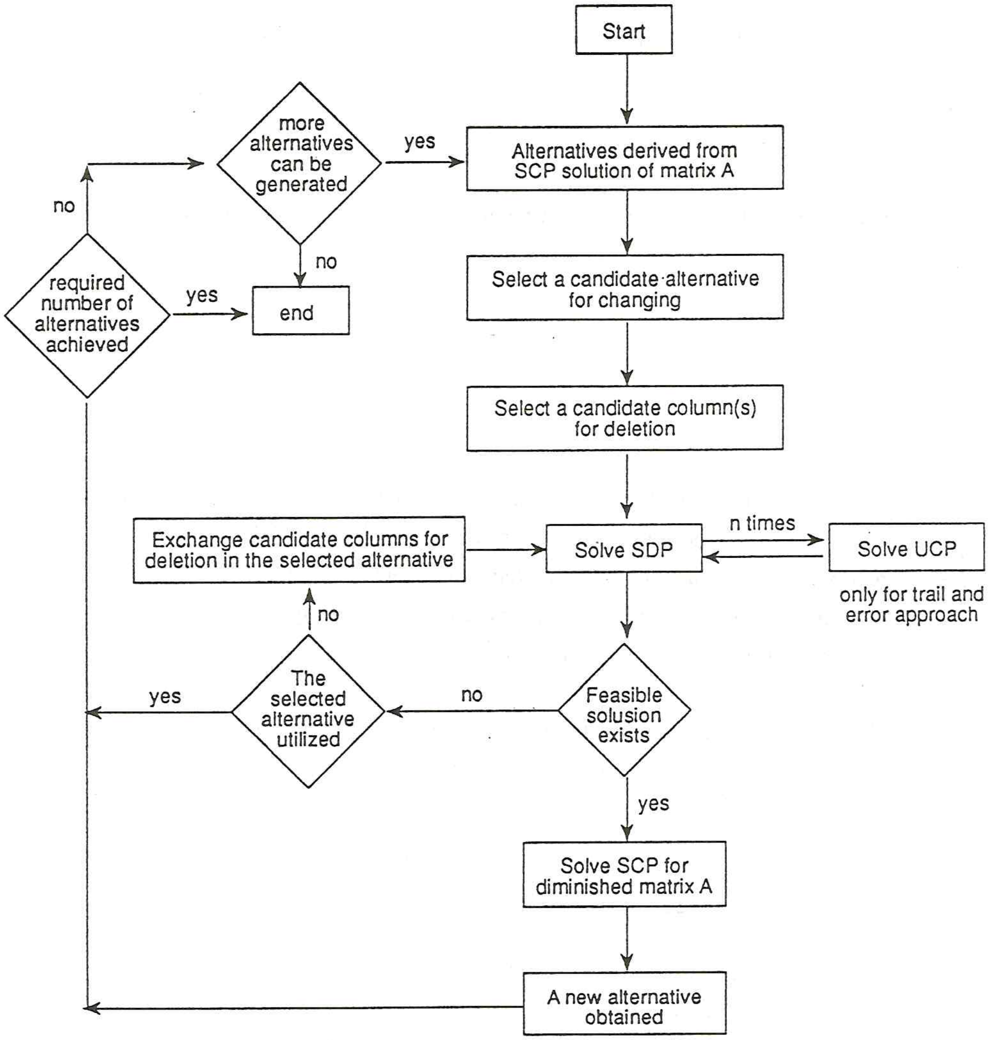
1. A feasible solution exists -- only $j \varepsilon J$ columns are in the solution;
2. A non-feasible solution exists -- the solution also comprises $f \varepsilon F$ columns.

The first possibility yields a covering to matrix A, while the latter does not. The linkage between these solutions and the master problem is expressed in step 11 of the master problem.

The entire process of the "alternative generation" is presented in a flow chart of Fig. 2.

3.4 Evaluation and Selection of Alternatives

The evaluation of the generated alternatives and the selection of "the best" is performed by the method of "compromised programming." This procedure, developed by Zeleny (1973, 1974), corresponds mainly to the solution of linear multi-objective problems (continuous), while a derivative variation of this method is used for solving discrete problems. One of the greatest benefits of this method is that it decreases the number of points to be checked in the Non-Inferior Solution set (NIS) and in concentrating on the section that lies on the NIS curve which provides the compromise solutions. In this paper, the general method will be explained first (the continuous), and following, its derivative for a discrete case which is utilized for this study. The compromise programming method -- defines the solutions which are the nearest to the ideal solution by calculating the index of the geometrical distance. These solutions are called the "compromise solutions," and they establish the "compromise set". The ideal solution is the one which yields all the objective functions to an optimum value simultaneously. Usually this is impossible; otherwise there would have been no conflict in the objective functions.


Start
Alternatives derived from SCP solution of matrix A
Select a candidate alternative for changing
Select a candidate column(s) for deletion
Solve SDP
n times
Solve UCP
only for trail and error approach
Feasible solusion exists
yes
Solve SCP for diminished matrix A
A new alternative obtained
no
The selected alternative utilized
no
Exchange candidate columns for deletion in the selected alternative
yes
required number of alternatives achieved
yes
end
no
more alternatives can be generated
yes
no

SCP = Set Covering Problem

SDP = Set Deletion Problem

UCP = Unique Combination Problem

Fig. 2: Alternative Generation Process -- a Flow Chart

The ideal solution is brought about as a theoretical point in the objective functions' space

$$Z^* = \left(Z_1^*, Z_2^*, \ldots, Z_p^*\right)$$

which results from solving a single objective optimum problem separately for each function.

$$\begin{aligned} &\min(Z_1(x)) \\ &s.t \\ &x \in X \\ &i = 1,2,\ldots,p \end{aligned} \qquad (3.20)$$

The ideal point which lies beyond the feasible range of the convex curve in the objective function space is used as a measure for evaluating the NIS solution. If possible, the entire objectives would be to attain the ideal point so that the values of the benefit function will have grown. Hence, we can conclude that acquiring solutions which are close to the ideal can be a substitute for maximizing a benefit function.

The measure used for evaluating the proximity between the NIS solutions and the ideal point is based on a family of metric distances, Ls, with the power of s value, as defined:

$$L_s = \left[\sum_{i=1}^{p}\left[\frac{Z_i(x) - Z_i^*}{Z_i^M - Z_i^*}\right]^S\right]^{1/s} \qquad ,0 \le s < \infty \qquad (3.21)$$

when:

$$Z_i^M = \max_{x \in X} \quad Z_i(x) \qquad ,i = 1,\ldots,p \qquad (3.22)$$

The vector x* of the solutions is the one to resolve the minimization problem:

$$\begin{aligned}&\min \quad L_s(x) = L_s(x_s^*)\\ &\text{s.t.}\\ &x \in X\end{aligned} \qquad (3.23)$$

The comprmise set is a set of the calculated compromise solutions (3.23) for each $o < s < \infty$ In actual fact, usually only the values $s = 1,2,\infty$ are used (Goichoechea et al. (1982)), and the decision-maker must decide which one.

Yu (1973) demonstrated that solving the problem will always produce an efficient point $\left(Z_i(x_s^*), i = 1,\ldots,p\right)$ on NIS to $1<s<\infty$, while for $s = \infty$ there will always be at least one efficient point. Zeleny (1973, 1974) named such solutions as the "compromise set" remarks that all other compromise solutions X_s* of (3.23) will always be on NIS and will take place between the compromise solutions L_1 to L_∞ which means this method permits concentrating on the same section of NIS, providing the foremost solutions. As a result, the production and selection process of the entire NIS is decreased beforehand. Thus, determining different s values by the decision-maker is a type of productive method for the compromise curve.

Application of the compromise-programming method for solving discrete problems is according to Duckstein and Opricovic's (198O) method. The continuous type formulas are changed into discrete ones as follows, while in the study, the value p=4 is considered (four objective functions). From the process of alternative generation, a final number of feasible points (Z_1, Z_2, Z_3, Z_4) are attained, and can be ordered in a pay-off table. Hence, the values are calculated as follows:

$$Z_i^* = \min_{k \in SOL} Z_i^k \qquad \forall i = 1,\ldots,4 \qquad (3.24)$$

$$Z_i^M = \max_{k \in SOL} Z_i^k \qquad \forall i = 1,\ldots,4 \qquad (3.25)$$

$$L_s^k = \left[\sum_{i=1}^{4} \left[\frac{Z_i^k - Z_i^*}{Z_i^M - Z_i^*} \right]^s \right]^{1/s} \qquad \forall \quad k \in SOL, \quad 1 \le s < \infty \qquad (3.26)$$

The compromise solutions are those which determine minimal distances from the ideal solutions to each given s:

$$L_s^* = \min_{k \in SOL} L_s^k \qquad \forall \quad 1 \le s < \infty \qquad (3.27)$$

In the research, values $s = 1,2, \infty$ have been considered.

4. NUMERICAL EXPERIENCE

The idea behind the entire heuristic process of alternative generation is to provide most of the efficient points during the first stages of the process. Executing more stages yields the production of more inefficient points. The sufficient number of alternatives is due to the structure of the matrix; otherwise, all the alternatives will be provided as in a complete enumeration procedure.

The initial condition for choosing a candidate set was given in (3.5) - step 5 of the master problem. In order to examine this condition, it was compared to other initial conditions on an empirical basis. Two groups of 5O small size (1O nodes) networks in each were produced. The first group was designed in such a fashion that it might produce a high number of alternatives from each network (depending on density, number of terminals, their location, etc.). The second group was designed to produce a small number (less than 1O) alternatives (usually small or low density networks). The initial conditions that were compared:

(1) The set with the minimal "potential metric distance":

$$\min_{k \in CAN} \left(L_s^k + \Delta L_s^k \right)$$

(2) The set with the minimal metric distance:

$$\min_{k \in CAN} L_s^k$$

(3) the set with the maximal metric distance:

$$\max_{k \in CAN} L_s^k.$$

The first group of problems examine the solution quality. The results are described in Table 3. "Accuracy level" defines the relative times the solution provided was identical to the minimal one (minimal metric distance) that was yielded by the approach. The left part of the table describes those cases that did not provide the minimal solution. The frequencies distribution (in percent) denote the closeness to the minimal solution (for example -- in approach (1), 50% of the alternatives that do not provide minimal value to the criterion have a distance of not more than 20% from the minimal value that was created).

Table 3: The Influence of Different Initial Conditions in the Master Problem on the Solution

Approach No.	Acc. level	Closeness to minimal value (%)				
		0-20%	20-40%	40-60%	60-80%	80-100%
(1)	88	50	33	17	0	0
(2)	60	25	25	45	4	1
(3)	28	6	14	28	20	32

The second group of problems examine the location of the efficient alternatives during the process. Table 4 presents the frequencies' distribution (in percent) of the alternatives that achieved the minimal value, due to the process stages (as, for example, in approach (1), 65% of the alternatives with the minimal value were generated during the production of 25% of the alternative of a certain network).

Table 4: The Influence of Different Initial Conditions in the Master Problem on the Location of the Minimal Solution

Approach No.	Location distribution of solution			
	O-25%	25-5O%	5O-75%	75-1OO%
(1)	65	20	10	5
(2)	40	25	25	10
(3)	5	20	30	45

It can be concluded from these tests that approach (1) which was used for the initial conditions of the master problem is effective.

As to tcsting the algorithm with different size networks, the heuristic approach's results were compared to those provided from a complete enumeration. Random networks were produced of 4-15 nodes with different densities and with a various number of terminals (and their location in the network). Each category contains 8O networks, from each 12 solution alternatives (defining SOL) were provided (if this was possible). Table 5 presents the results due to the criterion of min L_S (weighted by different s values). "Total accuracy level" denotes the number of times (in percents) the solution provided (from one of the 12 alternatives) was equal to the minimal value. "Solution location distribution" denotes the number of times the minimal solution was obtained in the first set. "Max-min relative error" is an index which examines the alternatives in SOL which do not provide the optimal solution. The error is calculated as follows: from group SOL, which comprises the 12 alternatives that do not contain the optimal solution, the alternative with the minimal L_S was selected. Then the relative error δ from the global minimum is calculated. The alternative with the "max min δ" is selected from all the groups {SOL}, which means maximal relative error. The "portion of efficient solutions" provides the portion (in percents) of efficient solutions (NIS) in the generated alternatives out of the total number of efficient solutions derived from the complete enumeration.

The table shows an "average accuracy level" of 9O%. This value decreases while the network dimension increases due to the portion between the 12 alternatives and the increasing number of enumerated alternatives (which might yield efficient alternatives not contained in the solution). The "max-min relative error" shows a relatively small value even for the worst alternative, and a total average of 9.8%. The "portion of efficient solutions" denotes decreasing values due to the increase in problem size ensuing from the same reason as explained earlier.. The worst case shows a relatively high percentage -- 8O% of efficient solutions.

Table 5: Efficiency of Heuristic Algorithm in Achieving Minimal L_S

Approach No.	Total Acc. level	Solution location distribution				max-min δ	portion of efficient solutions
		O-25%	5-5O%	5O-75%	75-100%		
4-6	100	84	16	0	0	0.0	100
7-9	91	64	17	9	1	8.4	94
10-12	86	54	15	12	5	12.6	88
13-15	82	50	18	7	7	18.1	80
average	90	63	17	7	3	9.8	90.5

5. SUMMARY AND CONCLUSIONS

The heuristic approach described in this paper provides: (a) generation of finite sets of alternatives efficient non-inferior solutions; (b) evaluation and selection of the various solutions using multi-objective preference techniques for discrete variables ("compromise programming" procedure).

The "alternative generation process" is based on a given "travel paths matrix" which can be mapped by several techniques of "column generation" in previous stages. This matrix consists of all possible paths (direct routes and via transfers), subject to certain constraints, versus all origin-destination pairs of the network.

The iterative procedure is constructed in a modular form, successively confronting the NP-hard type complexity problem, and its formulation as non-linear (concave) programming with mixed variables (continuous and integer). Such a formulation cannot be solved via known mathematical programming approaches and packages.

The proposed approach is compared with a mathematical programming optimal solution for small networks, and by enumeration optimal solution for medium size networks. The two comparison measures related to that are the quality of solution and computational time. These tests show that the heuristics used are efficient in an overall perspective.

The proposed approach is believed to be a useful toolset for the following applications:

(a) Optimal design for a new transit network;

(b) Optimal design for expansion or curtailment of an existing transit network;

(c) Assessment of the performance of an existing transit network from the aspects of: (i) Operator efficiency; (ii) Passenger level of service;

(d) Sensitivity analysis of transit network performance for a variety of system parameters (such as different bus fleet size, different level of service), changes in passenger demand, changes in frequencies, changes in travel time and more).

REFERENCES

Ceder, A., Israeli, Y. (1992). "Scheduling Considerations in Designing Transit Routes at the Network Level." Lecture Notes in Economics Mathematical System," No. 386, Proc. of the 5th Int. Workshop on Computer-Aided Transit Scheduling, Montreal, Canada, August 199O, by M. Descrochers and J.M. Rousseau (Eds.), Springer-Verlag, pp. 113-136.

Ceder, A., Stern, H. (1981). "Deficit Function Bus Scheduling with Deadheading Trip Insertion for Fleet Size Reduction." Trans. Science, Vol. 15, No. 4, Nov., pp. 338-363.

Ceder, A., Wilson, N. (1986). "Bus Network Design." Trans. Res., Vol. 2O8B, No. 4, pp. 331-334.

Cohon, J.L. (1978). "Multi-Objective Programming and Planning." Academic Press, New York.

Current, J., Min, H. (1986). "Multi-Objective Design of Transportation Network: Taxonomy and Annotation." European Journal of Operational Research, Vol. 26, pp. 187-2O1.

Current, J., Revelle, C.S., Cohon, J.L. (1987). "The Median Shortest Path Problem: A Multi-objective Approach to Analyze Cost vs. Accessibility in the Design of Transportation Networks." Transportation Science, Vol. 21, No. 3, August, pp. 188-197.

Duckstein, L., Opricovic, S. (198O). "Multi-Objective Optimization in River Basin Development." Water Research, Vol. 16, No. 1, pp. 14-2O.

Fandel, G., Spronk, J. (Eds.), (1985). "Multiple Criteria Decision Methods and Application: Selected Reading of the 1st International Summer School, Acireale, Sicily, September 1983." Springer, Berlin.

Flynn, J., Ratick, S. (1988). "A Multi-objective Hierarchical Model for the Essential Air Services Program." Trans. Science, Vol. 22, No. 2, May, pp. 139-147.

Fricker, J.D., Shanteau, R.M. (1986). "Improved Service Strategies for Small City Transit." Trans. Res. Record, No. 1O51, pp. 3O-34.

Gabbani, D., Magazine, M. (1986). "An Interactive Heuristic Approach for Multi-Objective Integer Programming Problems." Journal of Operations Research Society, Vol. 37, pp. 285-291.

Goicoechea, A., Hansen, D.R., Duckstein, L. (1982). "Multi-Objective Decision Analysis with Engineering and Business Applications." John Wiley & Sons, N.Y.

Hwang, C., Masud, A. (1979). "Multiple Objective Decision-Making -- Methods and Applications." Springer, N.Y.

Ignizio, J.P. (1983). "An Approach to the Modeling and Analysis of Multi-objective Generalized Networks." European Journal of Operational Research, Vol. 12, pp. 357-361.

Israeli, Y. (1992). "Transit Route and Scheduling Design at the Network Level." Doctoral Dissertation, Civil Engineering Department, Technion-Israel Institute of Technology, Israel.

Janarthanan, N., Schneider, J. (1986). "Multi-Criteria Evaluation of Alternative Transit System Designs." Trans. Res. Record, No. 1O64, pp. 26-34.

Lazimy R. (1986). "Interactive Relaxation Method for a Broad Class of Integer and Continuous Non-linear Multiple Criteria Problems." J. Math. Anal. Appl., Vol. 116, pp. 553-573.

Lee, S., Moore, L. (1977). "Multi-Criteria School Busing Models." Management Science, Vol. 23, pp. 7O3-715.

Magnanti, T.L., Wong, R.L. (1984). "Network Design and Transportation Planning: Models and Algorithms." Trans. Sci., Vol. 18, No. 1, February, pp. 1-55.

Minieka, E. (1978). "Optimization Algorithms for Networks and Graphs." Marcel Dekker Inc., N.Y., pp. 181-234.

Pogun, G., Satir, A. (1986). "Alternative Bus Scheduling Policies for an Exclusive Bus Lane." Trans. Res., Vol. 2OA, No. 6, pp. 437-446.

Rasmussen, L.M. (1986). "Zero-One Programming with Multiple Criteria." European Journal of Operational Research, Vol. 26, pp. 83-95.

Steenbrink, P.A. (1974). "Optimization of Transport Networks." Offset drukherij N.V., Nederlandse Spoorwegen, Utrecht, Holland/John Wiley & Sons Ltd.

Syslo, M.M., Deo, N., Kowalit, J.S. (1983). "Discrete Optimization Algorithms." Prentice-Hall Inc., New Jersey, Chapter 2, pp. 176-211.

Tadi, R.R., Khasnabiss, S.,Opiela, K.S. (1986). "A Methodology for Evaluating Bus Service Cutback Programs." Transportation Quarterly, Vol. 4O, No. 2, April, pp. 243-261.

Teodorovic, D., Krcmar-Nozic, E. (1989). "Multi-Criteria Model to Determine Flight Frequencies on an Airline Network under Competitive Conditions." Trans. Sci., Vol. 23, No. 1, Feb., pp. 14-25.

Tzeng, G.H., Shiau, T.A. (1988). "Multiple Objective Programming for Bus Operations: A Case Study for Taipei City." Trans. Res., Vol. 22B, No. 3, pp. 195-2O6.

Yu, P. (1973). "A Class of Decisions for Group Decision Problems." Management Science, Vol. 19, No. 936.

Zeleny, M. (1973). "Compromise Programming: Multiple Criteria Decision-Making," J.L. Cohon and M. Zeleny (Eds.). University of S. Carolina Press, Columbia.

Zeleny, M. (1974). "A Concept of Compromise Solutions and the Method of the Displaced Ideal." Computers and Operations Research, Vol. 1, No. 4, pp. 479-496.

Path Location Models for Bus Network Design

Pasquale Avella (*+)

Antonio Sforza (*#)

Abstract

The Bus Network Design Problem consists of determining a set of bus lines which optimizes given functions of the user costs and of the management costs, satisfying the transportation demand and the resource constraints.

Particularly, the design of a single line can be viewed as the problem of locating a primary path on a network, having some additional properties. It can be formalized as a Median Path problem, that is the problem of determining a primary path on a network minimizing a function of the accessibility to the same path and of the total cost of the path.

Accessibility is expressed by the total weighted distance travelled by users on secondary paths to reach the primary path from every node of the network. Total cost of the path is expressed by the sum of the costs of the arcs constituting the primary path.

For this problem we describe the formulation given by Current, ReVelle and Cohon (1987), with a number of constraint exponentially growing with the number *n* of the network nodes, and propose a new formulation, based on a graph transformation, containing a number of constraints polynomial in *n*. This compact formulation allows to solve the problem by an integer linear programming optimizer. Moreover it can be suitably relaxed to generate lower and upper bounds for the optimal solution by a lagrangean heuristic.

This work is supported by 2nd Special Project on Transportation of the National Research Council of Italy

An extended version of this paper will be submitted to Transportation Science

(*) Unità Operativa Progetto Finalizzato Trasporti 2 CNR

c/o Dipartimento di Ingegneria dell'Informazione e Matematica Applicata, Università di Salerno,

Via Ponte Don Melillo, 84084 Fisciano (Sa)

(#) Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II",

Via Claudio 21, 80125 Napoli

(+) Dottorato di ricerca in Ricerca Operativa, Università di Roma "La Sapienza"

1. Introduction

The planning process of a mass transit system can be divided in five main steps: bus network design, computation of frequencies, timetabling, bus scheduling, drivers scheduling. The first three steps concern with strategic and tactical management, the last two ones with operational management.

The Bus Network Design Problem consists of determining a set of lines which optimizes given functions of the user costs and of the management costs, satisfying the transportation demand and the resource constraints. Two objective functions have to be minimized:

- the management costs of the lines, on the side of the Bus Company;
- the transportation costs, on the side of the users.

Several heuristic methods have been proposed to solve this problem, based on a detailed description of the different parameters involved. For a good survey of these methods, see Ceder and Wilson (1985) and List (1990). The limit common to the most of them is the absence of a theoretical model and of an analysis of the computational results. Moreover they are presented as multi-line methods, but really they solve the problem line by line.

To design a single line a different solution approach could be adopted, related to the "path finding" problems, with a reduced set of parameters. They consist of determining a path on a network with some specific properties. Among this class of problems we can mention the k-shortest path and the node constrained shortest path, with or without time windows at the nodes (Ibaraki, 1973; Deo and Pang, 1984; Desrochers and Soumis, 1985; Skiscim and Golden, 1989).

In the case of a single line a very interesting and promising solution approach can be adopted , based on the interpretation of the Bus Line Design problem as a Path Location problem. This term is adopted in the field of network location when the facility to be located is a path-shaped facility, as in the case of a bus line (Avella e Sforza, 1993).

In the field of path location problems the median path problem assumes particular interest. It is the problem of determining a primary path on a network minimizing a function of the accessibility to the same path and of the total cost of the path.

Accessibility is expressed by the total weighted distance travelled by users on secondary paths to reach the primary path from every node of the network, that is the sum of the distances from the primary path to all the nodes not belonging to it, also defined as the "distance" of the path.

Total cost of the path is expressed by the sum of the costs of the arcs constituting the primary path.

Therefore two criterion can be assumed to locate a median path:

a) the distance of the path (to minimize);

b) the total cost of the path (to minimize).

It is useful distinguish the cases of tree and cyclic network.

For a tree there is only one path for each pair of nodes, and so the median path is defined without fixing sink and source. It is defined as the path P whith minimum "distance", with or without an upper bound constraint (a budget constraint) on its total cost. For this problem we find the works of Morgan and Slater (1980), Slater (1982), Minieka e Patel (1983), Minieka (1985). In particular Morgan and Slater (1980) proposed a very efficient solution algorithm with complexity O(n), if n is the number of network nodes, based on a particular data structure of the tree.

For a cyclic network we could define the median path problem as a bi-objective problem, with two conflicting objective functions. Indeed, in the most of cases, the total cost of the path has to be increased to reduce the distance of the path, that is the total weighted distance travelled by users on secondary paths to reach the primary path from every node of the network. Current et al. (1987) proposed a bi-objective model for the median path problem with a source s and sink t and an heuristic approach to its solution based on the k-shortest paths between s and t, computed with criterion (b) and compared with criterion (a).

In the case of cyclic networks the bi-objective problem can be reduced to a single objective problem in two different ways.

The first consists in transforming the function (b) in a budget constraint on the maximum cost of the path. The second consists in determining a convex combination of the two objective functions (a) and (b). Each of the two objective functions may be suitably weighted, to satisfy all the possible preferences of the decision-maker. So we obtain two different single objective problems, which can be named, respectively, Budget Median Path and Fixed Charge Median Path.

The Budget Median Path Problem on general networks was studied by Richey (1990) and Hakimi et al. (1993) for the aspects of theoretical computational complexity. Richey found that the Budget Median Path Problem is NP-hard on general networks and polynomial on series-parallel networks.

For the Fixed Charge Median Path Problem neither theoretical nor empirical results are avalaible in literature. However it could be heuristically solved through the algorithm proposed by Current et al. (1987).

In the section 2 of the paper the median path problem is formally defined and the model formulation proposed by Current et al. (1987) is reported, with a simple extension to the case of median path with no fixed source and sink. In the section 3 a tranformation of the graph is described. This transformation permit to interpret a solution of the problem as a special Steiner arborescence and to develop a new formulation, containing a number of constraint which is polynomial in the number of arcs. This compact formulation allows to solve the problem by an integer linear programming optimizer. Moreover it can be suitably relaxed to generate lower and upper bounds for the optimal solution by a lagrangean heuristic. The work perspective consist in developing a dual based approach for its solution. An other perspective is the formulation of a median path model on a network with origin/destination transportation demand.

References

Avella P. e Sforza A. (1993), "Modelli di Path Location per il Bus Network Design", Atti del I Convegno del Progetto Finalizzato Trasporti 2 - C.N.R., Roma, ottobre 1993 (in italian).

Ceder A. and Wilson N.H.M. (1985), "Bus Network Design", Transportation Research B, 20B, 4, 331-344.

Current J.R., ReVelle C.S. e Cohon J.L. (1987), The Median Shortest Path Problem: A multiobjective approach to analyze cost Vs. accessibility in the design of transportation networks, Transportation Science 21, 188-197.

Deo N. and Pang C. (1984), "Shortest-Path Algorithms: taxonomy and annotations", Networks, 14, 275-323.

Desrochers M. and Soumis F. (1985), "A generalized permanent labelling algorithm for the shortest path problem with time windows", C.R.T. Université de Montreal, Publication #394A, 28 pp.

Hakimi S. L., Schmeichel E.F. and Labbè M. (1993), "On locating path - or tree- shaped facilities on networks. Networks 23, 1993.

Harary F. (1969), "Graph Theory", Addison Wesley, Reading, Massachusetts.

Ibaraki T. (1973), "Algorithms for obtaining shortest paths visiting specified nodes", SIAM Review 15, 309-317.

Minieka E. (1985), "The Optimal Location of a Path or Tree in a Tree Network", Networks 15, 309-321.

List G.F. (1990), "Toward Optimal Sketch-Level Transit Service Plans", Transportation Research B, 24B, 5, 325-344.

Minieka E. and Patel N.H. (1983), "On Finding the Core of a Tree with a Specified Lenght", Journal of Algorithms 4, 345-352.

Morgan C.A. and Slater P.J. (1980), "A Linear Algorithm for a Core of a Tree", Journal of Algorithms 1, 247-258.

Richey M.B. (1990), "Optimal Location of a Path or Tree on a Network with Cycles", Networks 20, 391-407.

Skiscim C.C. and Golden B.L. (1989), "Solving k-Shortest Path and Constrained Shortest Path Problems Efficiently", Annals of Operations Research, 20, 249-282.

Slater P.J. (1982), "Locating Central Paths in a Graph", Transportation Science, 16, 1-18.

Tansel B.C., Francis R.L.e Lowe T.J. (1983), "Location on networks: A survey", Management Science 29, 482-511.

A PILOT ALGORITHM FOR LARGE-SCALE NONLINEAR NETWORK OPTIMIZATION

Lucio Grandinetti, Francesca Guerriero, Roberto Musmanno

Dipartimento di Elettronica, Informatica e Sistemistica,
Università della Calabria, Rende (CS) - Italy

Abstract. This paper deals with a pilot algorithm designed to solve very large-scale network flow optimization problems.

The main aim is to emphasize the use of modern information technologies like high performance parallel computers to obtain powerful computational tools for hard-to-solve network problems which arise in several applications, including transportation.

In this paper we discuss some numerical aspects of the dual relaxation algorithm on shared-memory parallel architectures, to focus on the effectiveness of the parallel approach especially when applied to large-scale transportation problems, based on a dynamic allocation of the nodes to processors, on the basis of which each node is scheduled to a different processor at the beginning of each iteration.

We present and analyze computational results which demonstrate that asynchronous implementations always outperform the synchronous and excellent speedup factors may be achieved for the majority of test problems.

Keywords. Nonlinear network optimization, relaxation method, complementary slackness conditions, shared memory multiprocessor, dynamic allocation.

1. INTRODUCTION

In this paper we consider the problem of solving a particular class of nonlinear network flow problems which are involved in several kind of applications, including transportation. Recent developments in network optimization take into account the idea of parallel computing in the design, analysis and implementation of new strategies for solving large-scale problems. As we will show in the sequel, the structure of the network problems plays a crucial role in a parallel computation context, and it may be amenable for both synchronous and asynchronous implementations.

We choose to focus on pure network problem with a special case of the objective function, assumed here separable. Given a directed network $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$, where $\mathcal{N} = \{i \mid i = 1, 2, \ldots, n\}$ is the set of nodes and $\mathcal{A} = \{(i,j) \mid i, j \in \mathcal{N}\}$ is the set of arcs, the problem (P) can be formally defined as follows:

$$\textit{minimize} \quad \sum_{(i,j)\in\mathcal{A}} c_{ij}(x_{ij}) \tag{1}$$

s.t.

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:(j,i)\in\mathcal{A}} x_{ji} = b_i \qquad , \forall\, i \in \mathcal{N} \tag{2}$$

$$l_{ij} \le x_{ij} \le u_{ij} \qquad , \forall\, (i,j) \in \mathcal{A} \ , \tag{3}$$

where x_{ij} is the flow on the arc (i,j), $c_{ij}: \mathcal{A} \longrightarrow \mathcal{R}$ is a strictly convex cost function of one variable (assumed to be differentiable), b_i is the supply at node i, l_{ij}, u_{ij} are respectively the lower and upper bound for x_{ij} on the arc (i,j); therefore, (2) e (3) specify conservation of flow constraints and capacity constraints.

An iterative method to solve (P), based on the solution of the dual problem, is considered. In section 2 we present the methodological ideas for the design of the relaxation method. In section 3 we explain how the relaxation method can be usefully implemented on parallel computers. Finally, numerical results, collected using an Alliant with 8 processors on significant test problems, are reported and discussed in section 4.

2. THE RELAXATION METHOD

A dual problem of (P) can be formulated as follows. We assume primarily that (P) is always feasible, that is:

$$\sum_{i \in \mathcal{N}} b_i = 0 \,. \tag{4}$$

This means that there exists a unique optimal solution, given the hypothesis on the cost functions c_{ij} and the fact that the region defined by the constraints (2)-(3) forms a compact set.

Attaching a vector of lagrangian multipliers $\pi \in \Re^n$ to the equality constraints (2), we obtain the following lagrangian function:

$$\begin{aligned} \mathcal{L}(x,\pi) &= \sum_{(i,j)\in\mathcal{A}} c_{ij}(x_{ij}) + \sum_{i\in\mathcal{N}} \pi_i \Big(\sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} - \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} + b_i \Big) \\ &= \sum_{(i,j)\in\mathcal{A}} (c_{ij}\,(x_{ij}) + (\pi_j - \pi_i)x_{ij}) + \sum_{i\in\mathcal{N}} \pi_i b_i \,. \end{aligned} \tag{5}$$

Assigned $\pi \in \Re^n$, the value of the dual function $q(\pi)$ can be obtained by minimizing $\mathcal{L}(x,\pi)$ over all flow distributions x which satisfy the lower and upper bounds. Therefore, the dual (D) of (P) can be formulated as follows:

$$\mathop{\mathit{maximize}}_{\pi \in \Re^n} \quad q(\pi) \,, \tag{6}$$

where:

$$q(\pi) = \min_{l_{ij} \le x_{ij} \le u_{ij}} \quad \mathcal{L}(x,\pi) = \sum_{(i,j)\in\mathcal{A}} q_{ij}\,(\pi_i - \pi_j) + \sum_{i\in\mathcal{N}} \pi_i b_i \,, \tag{7}$$

and

$$q_{ij}\,(\pi_i - \pi_j) = \min_{l_{ij} \le x_{ij} \le u_{ij}} \quad c_{ij}\,(x_{ij}) - (\pi_i - \pi_j)x_{ij} \,. \tag{8}$$

A pair (x,π) is optimal respectively for the primal and the dual problem if and only if the following conditions, referred as complementary slackness, are satisfied.

$$t_{ij} \ge \frac{\partial c_{ij}(x_{ij})}{\partial x_{ij}}\bigg|_{x_{ij} = u_{ij}} \qquad \Rightarrow \qquad x_{ij} = u_{ij}\,; \tag{9a}$$

$$t_{ij} \le \frac{\partial c_{ij}(x_{ij})}{\partial x_{ij}}\bigg|_{x_{ij} = l_{ij}} \qquad \Rightarrow \qquad x_{ij} = l_{ij}\,; \tag{9b}$$

$$t_{ij} = \frac{\partial c_{ij}(x_{ij})}{\partial x_{ij}}\bigg|_{x_{ij} = \bar{x}_{ij}} \,,\; l_{ij} \le \bar{x}_{ij} \le u_{ij} \qquad \Rightarrow \qquad x_{ij} = \bar{x}_{ij} \,, \tag{9c}$$

where we assume that $t_{ij} = \pi_i - \pi_j$.

Furthermore, if we consider the partial derivative $\frac{\partial q(\pi)}{\partial \pi_i}$ of the dual function q with respect to the i-th price π_i, we obtain:

$$\begin{aligned} \frac{\partial q(\pi)}{\partial \pi_i} &= \sum_{(m,n)\in A} \frac{\partial q_{mn}(\pi_m - \pi_n)}{\partial \pi_i} + b_i = \\ &= - \sum_{\{j:(j,i)\in A\}} \nabla q_{ji}(\pi_j - \pi_i) + \sum_{\{j:(i,j)\in A\}} \nabla q_{ij}(\pi_i - \pi_j) + b_i = \end{aligned}$$

$$= \sum_{\{j:(j,i)\in A\}} x_{ji} - \sum_{\{j:(i,j)\in A\}} x_{ij} + b_i , \tag{10}$$

that is, $\frac{\partial q(\pi)}{\partial \pi_i}$ corresponds to the surplus g_i of the node i associated with the unique flow distribution x for which the pair (x,π) satisfies the complementary slackness conditions.

Differentiability and convexity of q play a crucial role for the definition of iterative methods for solving (D), since, starting from a generic estimate of the dual solution we can improve the dual function along a direction defined by a single component π_i of π. The greatest improvement along the coordinate direction π_i corresponds to $\frac{\partial q(\pi)}{\partial \pi_i} = 0$, that is, when the surplus $g_i = 0$.

The sketch of k-th iteration of the relaxation algorithm for maximizing $q(\pi)$ is depicted in Fig. 1:

1. Assigned $\pi^{(k)}$ e computed $x^{(k)}$ by means of complementary slackness (9), *choose* a node $i \in \mathcal{N}$ such that $|g_i^{(k)}| > \varepsilon$, where ε is a tolerance defined by the user; if such node does not exist, STOP, $x^{(k)}$ is optimal.

2. *Compute* $\pi^{(k+1)}$ such that $\pi_j^{(k+1)} = \pi_j^{(k)}$, $\forall\ j \neq i$, with $\pi_i^{(k+1)}$ chosen in such a way $q(\pi)$ could be improved along the i-th coordinate.

3. Determine $x^{(k+1)}$, that is $x_{ij}^{(k+1)} = x_{ij}^{(k)}$ e $x_{ji}^{(k+1)} = x_{ji}^{(k)}$, if the node j is not adjacent to i, whereas for adjacent nodes j, $x_{ij}^{(k+1)}$ is chosen in such a way the complementary slackness conditions are satisfied:

$$\left.\frac{\partial c_{ij}(x_{ij})}{\partial x_{ij}}\right|_{x_{ij}=x_{ij}^{(k+1)}} = \pi_i^{(k+1)} - \pi_j^{(k+1)}.$$

4. Compute the surplus $g_j^{(k+1)}$ for j=i and for all nodes j adjacent to i according to the flow distribution $x^{(k+1)}$.

Fig. 1
k-th iteration of the relaxation method.

The asymptotic convergence of the method can be proven ([1]), starting from any estimate π of the solution π^*.

3. PARALLEL IMPLEMENTATIONS OF THE RELAXATION METHOD

The relaxation method is particularly suited for both synchronous and asynchronous parallel implementation, since the iterations could be performed for many nodes concurrently; in the synchronous case, the computation proceeds in rounds of parallel iterations. During each round, at most *p* nodes are considered (where *p* is the number of processors), from a list of candidate nodes for which the optimality conditions are not satisfied. The processors then execute in parallel the steps 2, 3, and 4 of the method depicted in Fig. 1 and a new round begins once all processors have finished. Such scheme works without altering the relaxation method only if we never iterate simultaneously on the dual variables of adjacent nodes. This involves the idea of a coloring of the network graph $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$ for partitioning the node set $\mathcal{N}$ into subsets such that any two nodes of the same color are not adjacent in the network. Since the computation of the dual variable π_i is totally independent on the values of any variables associated to nodes having the same color of i, it turns out that the synchronous scheme does not affect the course of the algorithm. Moreover, the synchronous approach works better on transportation problems, for which only two colors are required. The parallel synchronous method is depicted in the following figure, which shows the synchronization point at the end of the surplus updating procedure. We observe that, even though at each iteration we never select adjacent nodes, we may contemporarily update the surplus of a node j, as shown in Fig. 3. In this case we need a lock to the shared memory location of the surplus to ensure that no other processor can change simultaneously the value. One way to accomplish this is that processor i could locally determine the variation $\Delta x_{ij} = x_{ij}^{(k+1)} - x_{ij}^{(k)}$ of the flow along the arc (i,j) and update the surplus $g_j = g_j + \Delta x_{ij}$, by using a lock. When all processors complete this procedure, we will obtain $g_j^{(k+1)} = g_j$.

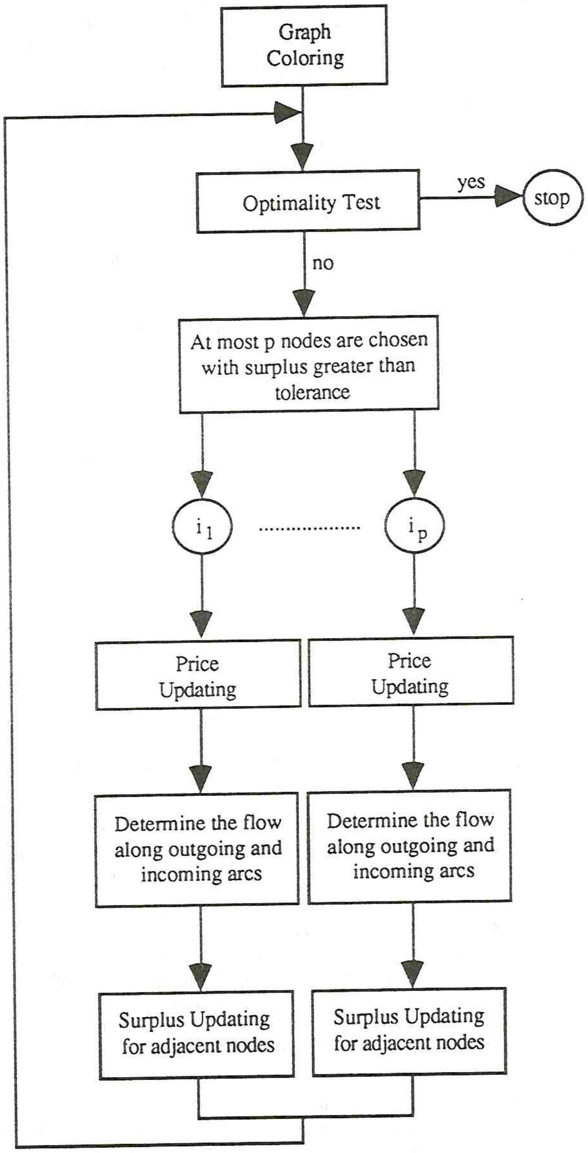

Graph Coloring
Optimality Test
yes
stop
no
At most p nodes are chosen with surplus greater than tolerance
i_1
....
i_p
Price Updating
Price Updating
Determine the flow along outgoing and incoming arcs
Determine the flow along outgoing and incoming arcs
Surplus Updating for adjacent nodes
Surplus Updating for adjacent nodes

Fig. 2

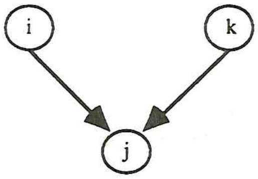


Fig. 3

Several variations of the procedure for selecting nodes at each iteration could be considered. In this paper two different extreme strategies are investigated:

a) cyclical node selection; the nodes are chosen in a cyclical order, by starting from the node selected at the previous iteration;
b) largest surplus selection; at each iteration, for each color ξ, the candidate nodes are the *p* nodes which correspond to the largest surplus, the sum of which is S_ξ; the selected node are chosen among the candidates by determining the color $\xi^*=\arg\max_{\xi}\{S_\xi\}$.

We refer to these two strategies as dynamic allocations of nodes to processors; there exists also an alternative procedure to assign a priori each node to a processor, using a procedure executed at the beginning of the first iteration; we have, in this case, a static allocation of nodes to processors. We found that, at least for transportation problems, the dynamic allocation outperforms the static allocation because of the reduced number of colors. In [2] the authors have shown that the static allocation is more preferable than dynamic for the particular class of test problems used in their experiments.

In an asynchronous algorithm there is no notion of rounds, and a new node can be considered from the candidate list by some processor while other processors are still executing on different nodes.

The convergence of the parallel asynchronous methods can be proven under weak assumptions. For more details see [1]. From the computational point of view the asynchronous methods seem to be more attractive, because they are not delayed by synchronizations requirements. In fact, we have found empirically that the asynchronous versions always outperform the synchronous counterpart. The main drawback is due to the need of a more sophisticated procedure at the end of the algorithm, through which we compute again the flow vector and the surplus for each node even if the optimality test is passed. This is due to the fact that during the steps of the asynchronous algorithms the information used by each processor in the computational phases could be outdated.

Another crucial computational aspect in the parallel relaxation method is, for each selected node i, the procedure for calculating the price π_i at k-th iteration. It could be implemented according to one of the following ideas:

a) maximization of q with respect to π_i ;
b) updating of π_i, by the relation: $\pi_i^{(k+1)}= \pi_i^{(k)}+ \delta$.

According to the first possibility, we have implemented two different procedures, on the basis of which the price $\pi_i^{(k+1)}$ is determined in such a way $g_i^{(k+1)}= 0$ (in the sequel we refer to these procedures as *Newton-like method* and *exact line search*; for a more detailed description of the two methods, see [2], [3], [4], and also [5]). The idea of the exact line search is that the surplus of node i is a non-differentiable, non-increasing function of the price π_i, as indicated in Fig. 4. The linear segments parallel to the π-axis correspond to the range of π_i for which all arcs are either active or inactive. The non-linear segments correspond to the range of value of π_i for which at least one arc is balanced.

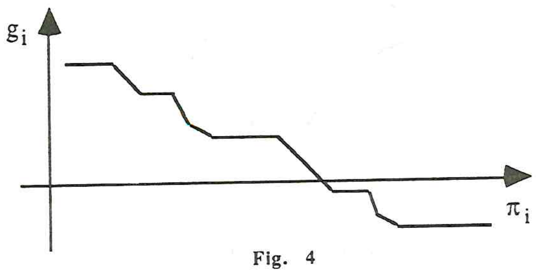


Fig. 4

We can easily implement a procedure to find the largest range of prices for which the surplus will change sign, by sorting the vector of the dual break points and using a logarithmic search to avoid the computation of the surplus in all break points. Thus, we can use a more efficient line-searching procedure in a tigther interval (the advantage of this approach is clear especially when we consider quadratic cost function for each arc; in this case it is trivial to find the price for which the surplus will be zero using a simple interpolation step between the extremes of the segment).

According to the second possibility, we have used similar approaches proposed in [2], and [3], by implementing the approximate line search of Tseng, the approximate Newton-like method, and the Bregman-step method. (we generally refer to these methods as *approximate line searches*). For a more detailed description of these approaches, we remind to [2], [6], and [7] .

4. COMPUTATIONAL EXPERIMENTS

Computational experiments have been carried out with the aim to conduct an accurate comparative analysis among versions of the algorithm which use different computational strategies for updating the dual prices and for scheduling the nodes to be processed. In particular, a set of 10 transportation problems are considered. The dimension of the problems is reported in the next table. The cost function c_{ij}, $\forall\ (i,j)\in \mathcal{A}$, is a quadratic-type. The problems are divided in two groups, each of them characterized by a fixed number of source and sink nodes. The number of arcs is selected so that the density of the networks is approximatively 0.05, 0.1, 0.15, 0.20, and 0.25. All the problems have been generated by using a modified version of the public domain software named Netgen ([8] and [9]).

Test	Nodes	Sources	Sinks	Arcs
1	1000	300	700	10500
2	1000	300	700	21000
3	1000	300	700	31500
4	1000	300	700	42000
5	1000	300	700	52500
6	10000	7000	3000	105000
7	10000	7000	3000	210000
8	10000	7000	3000	315000
9	10000	7000	3000	420000
10	10000	7000	3000	525000

Tab. 1
List of test problems.

We have used an Alliant FX/80, a shared memory multiprocessor system with 8 processors, each of them characterized by a peak performance of 23 Mflops and a core memory of 64 Mbytes. The compiler used is the FX/Fortran 4.2.40.

The computational results are summarized in the next two tables and figures, in which the results of the best sequential, synchronous, and asynchronous implementations of the relaxation method (in terms of execution time) are reported.

The tolerance ε is set to $\varepsilon = 10^{-3}\,\eta$, where $\eta = \frac{\sum_{i \in \mathcal{N}} b_i}{2m}$. The parameter η is a scaling which takes into account the difference between the demand and supply, which may substantially vary for each node;

First of all, we remark that all versions converged to a solution which satisfies the tolerance criterion. For the sequential versions, we observe that the cyclical node selection, for the first class of problems, is the most efficient, even if the number of iterations is remarkably increased. Such behavior may be explained observing that, since the cost per iteration is strictly dependent on the degree for each node, the computational workload is determined by the procedure for selecting the nodes, at least for low values of density. This means that it is preferable to implement a more sophisticated node selection strategy only when the computation of the prices will be more expensive, that is, for networks with higher values of density or for large-scale problems.

As a matter of fact, when we increase the number of nodes, the range of the density value for which the cyclical node selection strategy is more efficient, is reduced (with n=1000 the largest surplus node selection is cheaper starting with density of 0.01).

In the majority of cases, the most efficient line search procedure to compute the dual prices is the Newton-like method, a procedure which resembles the very-well-known method for nonlinear unconstrained optimization problems. This method guarantees a good balance between the need of a substantial reduction of the number of iterations (obtained with an exact line search) and the need to avoid to increase so much the computational cost per iteration.

However, the efficiency of the Newton-like method is mainly due to the particular type of the arc cost function considered in our computational experiments, assumed to be quadratic. In more general cases, we found that the exact line search procedure illustrated above is much more useful.

In the parallel case, we do not have a substantial reduction in the execution time for the synchronous versions with respect to the sequential counterpart (at least if we use as a term of comparison the best sequential implementation). This behavior can be seen in any experiment, that is, also when we increase the density of the networks, and there is not a big difference in terms of speedup using 8 processors instead of 4. However, the gap in the average cost per iteration between the two node selection strategies is relatively small, whereas in the sequential case the same difference is much more consistent. For several test problems it seems to be cheaper, in terms of execution time, using more sophisticated node selection strategies. This is essentially due to the availability of additional processors, which guarantee the possibility to use parallelism in an effective way, when the largest surplus must be computed at each iteration (this represents a remarkable computational workload especially for the sequential versions).

In our experiments we have considered also a numerical comparison between the static and dynamic allocation of the nodes. We have obtained that only with low density values the maximum speedup was achieved with the static allocation. For more significant test problems, with higher dimensions, the dynamic allocation procedure always outperforms the static counterpart. For this reason the results with the static allocation are not reported in this paper.

Also in the parallel case, the most efficient line search procedure is the Newton-like method. However, when the number of nodes and the density is increased, it seems to be more useful using the exact line search (this may be explained by the fact that the number of dual break points is increased and, as a consequence, the fact that we avoid to compute the surplus in all dual break points will be useful compared with the time spent for sorting the vector of the break points; for more details see [5])

The most important result obtained in our experiments is that the parallel asynchronous versions of the relaxation method are considerable faster that their synchronous conuterparts. As a matter of fact, the values of speedup achieved are excellent for the majority of the test problems used.

A final remark is that, when the density of the networks is increasing, the speedup is decreasing. This is due to the fact that, when the density of the problems overcomes some limit, the probability that we simultaneously process adjacent nodes becomes relatively high (if we do not fix any sort in the node selction strategy, as in our parallel asynchronous codes). In this case, since it is possible that more processors attempt to update the flow on the same arc, we have to consider the extra overhead due to the synchronization mechanism introduced for locking the shared memory location during the writing procedure.

However, numerical experiments carried out on sparse graphs have shown that the availability of 8 processors is extremely useful in terms of speedup achieved.

Test Problem	Sequential	Parallel			
		Synchronous		Asynchronous	
		4 Processors	8 Processors	4 Processors	8 Processors
1	31.75 / 18205	18.67 / 4594	13.25 / 2329	10.44 / 4901	6.21 / 2880
2	52.82 / 13638	28.54 / 3439	19.35 / 1742	16.51 / 3447	9.35 / 1987
3	112.74 / 22319	59.52 / 5743	38.47 / 2870	39.05 / 5884	20.65 / 3043
4	76.47 / 5423	27.37 / 21	19.18 / 21	19.24 / 1485	11.82 / 809
5	94.91 / 4946	32.45 / 23	25.8 / 23	23.17 / 1307	14.3 / 1073

Tab. 2

Time in secs / number of iterations required to solve the test problems 1, 2, 3, 4, and 5. The number of iterations in the parallel implementations is computed on the processor with the maximum workload.

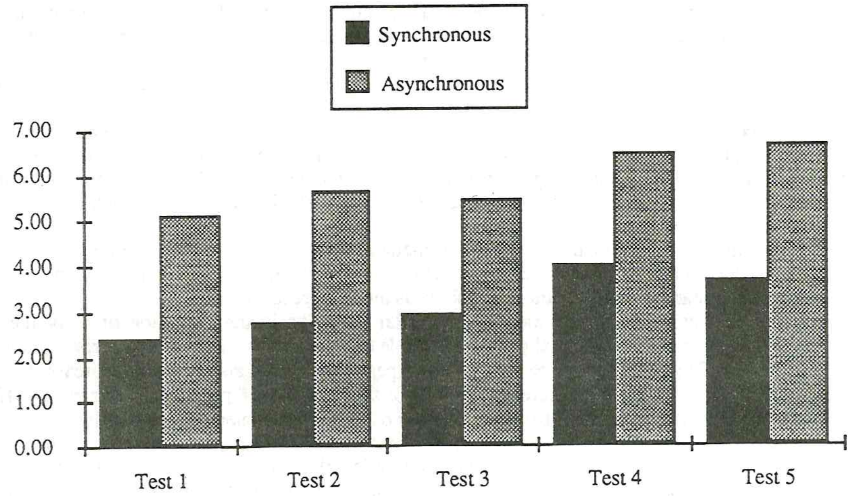


Fig. 5

Speedup values for the synchronous and asynchronous parallel codes (8 processors).

Test Problem	Sequential	Parallel			
		Synchronous		Asynchronous	
		4 Processors	8 Processors	4 Processors	8 Processors
6	600.77 / 137903	318.32 / 120	196.04 / 120	165.49 / 34487	94.62 / 17252
7	1158.2 / 91611	413.34 / 109	266.02 / 109	337.16 / 26075	232.12 / 14549
8	1596.5 / 79418	537.41 / 110	355.47 / 110	558.62 / 22630	322.38 / 11256
9	1906.8 / 68333	601.52 / 53	357.25 / 53	742.65 / 19628	432.79 / 10009
10	2298.5 / 64028	709.13 / 72	401.7 / 49	645.93 / 16517	457.9 / 9348

Tab. 3

Time in secs / number of iterations required to solve the test problems 6, 7, 8, 9, and 10. The number of iterations in the parallel implementations is computed on the processor with the maximum workload.

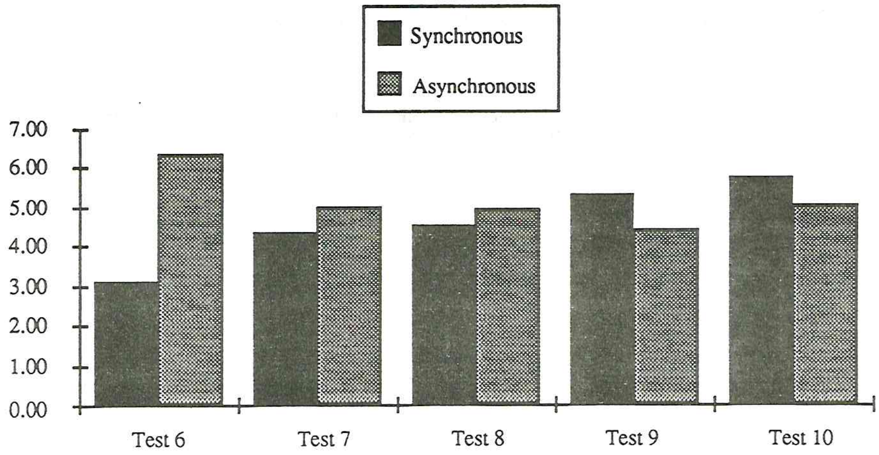


Fig. 6
Speedup values for the synchronous and asynchronous parallel codes (8 processors).

5. ACKNOWLEDGMENT

We would like to acknowledge the director and the staff of CERFACS, Toulouse, France, for the possibility to use the Alliant FX/80 for obtaining our computational results and Enrico Vena, for his fruitful collaboration given to us in collecting all the results.

6. REFERENCES

[1] Bertsekas D.P. and Tsitsiklis J.N., *Parallel and Distributed Computation (Numerical Methods)*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[2] Chajakis E.D. and Zenios A.S., "Synchronous and Asinchronous Impementations of Relaxation Algorithms for Nonlinear Network Optimization", *Parallel Computing*, (17), 873-894, 1991.

[3] Zenios S.A. and Nielsen S.S., "Massively Parallel Algorithms for Singly Constrained Nonlinear Programs", Report no. 90-03-01, Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, May 1990.

[4] Zenios S.A. and Mulvey J.M., A Distributed Algorithm for Convex Network Optimization Problems, *Parallel Computing* (6), 43-56, 1988.

[5] Zenios S.A. and Mulvey J.M., "Relaxation Techniques for Strictly Convex Network Problems", *Annals of Operations Research*, Vol. 5, pp. 517-538, 1985/6.

[6] Guerriero F., and Musmanno R., "Un algoritmo parallelo iterativo per il problema di flusso a costo minimo non lineare", Technical Report C.N.R. no. 1/164 - Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", Sottoprogetto 1: "Calcolo Scientifico per Grandi Sistemi", 1993.

[7] Grandinetti L., Guerriero F., and Musmanno R., "Parallel Techniques for Large-Scale Nonlinear Network Optimization", in: Lecture Notes in Computer Science (796), High Performance Computing and Networking, Wolfgang Gentzsch, Uwe Harms (Eds.), Springer Verlag, 1994.

[8] Klingman D., Napier A., and Stutz J., "NETGEN - A Program for Generating Large Scale

(Un)Capacitede Assignment, Transportation and Minimum Cost Flow Network Problems", *Management Science* (20), 814-822, 1974.

[9] Musmanno R., and Vena E., "GRICORN: un programma per la generazione di reti di flusso basato sul software di pubblico dominio Netgen", Technical Report C.N.R. no. 1/183 - Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", Sottoprogetto 1: "Calcolo Scientifico per Grandi Sistemi", April 1994.

A new class of primal-dual algorithms for multicommodity minimum convex cost network flow problems: serial and parallel implementations

By

Ismail Chabini and Michael Florian

Departement d'IRO et CRT
Université de Montréal
P.O. Box 6128, Station "A"
Montreal, Québec H3C 3J7, Canada

September 25, 1993

Extended abstract

The advent of parallel computing platforms has altered the implementation of network flow algorithms. Some algorithmic approaches, which did not appear to be efficient when the code development was designed for serial computers, become interesting for parallel computations. The algorithms presented in this paper are primal-dual methods which fully exploit the type of coarse grain parallelization possible on a network of Transputers, which is an MIMD computing platform. This methods are generalizations of solving procedures developped previously by the authors for the single commodity minimum convex cost network flow problems and for which very good results were obtained.

We consider first the strictly convex quadratic cost multicommodity transportation problem subject to upper bound contraints which are formulated as follows:

$$\min \ \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ijk}-c_{ijk})^2 \qquad (1)$$

$$
\begin{aligned}
\text{subject to:} -\sum_{j=1}^{m} x_{ijk} &= -O_{ik},\ i=1,2,\cdots,n;\ k=1,2,\cdots,K\ \{\alpha_{ik},\ \text{dual variables}\} && (2)\\
\sum_{i=1}^{n} x_{ijk} &= D_{jk},\ j=1,2,\cdots,m;\ k=1,2,\cdots,K\ \{\beta_{jk},\ \text{dual variables}\} && (3)\\
x_{ijk} &\le u_{ijk},\ \forall\,(i,j,k) \qquad \{\gamma_{ijk}\ ,\ \text{dual variables}\} && (4)\\
-x_{ijk} &\le 0,\ \forall\,(i,j,k) \qquad \{\delta_{ijk}\ ,\ \text{dual variables}\} && (5)\\
\sum_{k=1}^{K} x_{ijk} &\le U_{ij},\ \forall\,(i,j) \qquad \{\psi_{ij}\ ,\ \text{dual variables}\} && (6)
\end{aligned}
$$

We note that (2)-(6) define a bounded polytope which we assume to be nonempty. As stated above, (1) is strictly convex; hence, it follows that (1)-(6) admits a finite and unique solution.

The Lagrangian dual of (1)-(6) is

$$
\begin{aligned}
\max_{\psi\ge 0,\gamma\ge 0,\delta\ge 0,\alpha,\beta}\{\min_{x}\,[\,&\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ijk}-c_{ijk})^2+\sum_{k=1}^{K}\sum_{i=1}^{n}\alpha_{ik}(O_{ik}-\sum_{j=1}^{m}x_{ijk})+\\
&\sum_{k=1}^{K}\sum_{j=1}^{m}\beta_{jk}(-D_{jk}+\sum_{i=1}^{n}x_{ijk})+\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_{ijk}(x_{ijk}-u_{ijk})+\\
&\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{m}\delta_{ijk}(-x_{ijk})+\sum_{i=1}^{n}\sum_{j=1}^{m}\psi_{ij}(\sum_{k=1}^{K}x_{ijk}-U_{ij})]\} \qquad (7)
\end{aligned}
$$

The method, which we develop in this paper increases at each iteration the value of the dual function (7), by alternatively modifiying (α,β) , (γ,δ) and ψ. The algorithm is stated below:

Algorithm for Multicommodity Quadratic Cost Transportation Problem

Step 0 (Initialization) :

$t = 1$, $x^t = c$, $\overline{c}^t = c$, $T_k = \sum_{i=1}^{n} O_{ik} = \sum_{j=1}^{m} D_{jk}$, $q_{ijk} = (\frac{O_{ik}}{m} + \frac{D_{jk}}{n} - \frac{T_k}{nm}), \forall (i,j,k)$.
$\psi^t = 0, s^t = \delta^t - \gamma^t = 0$.

Step 1 (Projection on the flow conservation constraints) :

$$\overline{c}_{ijk}^{t+1} = \overline{c}_{ijk}^t + [\frac{\sum_{l,p} x_{lpk}^t}{nm} - \frac{\sum_p x_{ipk}^t}{m} - \frac{\sum_l x_{ljk}^t}{n} + q_{ijk}], \forall (i,j,k).$$

Step 2 (Projection on bounds) :

$$s_{ijk}^{t+1} = \max\{0, \min\{\overline{c}_{ijk}^{t+1} - \psi_{ij}^t, u_{ijk}\}\} - \{\overline{c}_{ijk}^{t+1} - \psi_{ij}^t\}, \forall (i,j,k).$$

Step 3 (Projection on the coupling constraints) :

$$\psi_{ij}^{t+1} = \max\{0, \tfrac{1}{K}(\textstyle\sum_{k=1}^{K}(\overline{c}_{ijk}^{t+1} + s_{ijk}^{t+1}) - U_{ij})\}$$
$$x_{ijk}^{t+1} = (\overline{c}_{ijk}^{t+1} + s_{ijk}^{t+1}) - \psi_{ij}^{t+1}.$$

Step 4 (Stopping criterion):

If $\max\{\max_{ijk}\{|\overline{c}_{ijk}^{t+1} - \overline{c}_{ijk}^t|, |s_{ijk}^{t+1} - s_{ijk}^t|\}, \max_{ij} |\psi_{ij}^{t+1} - \psi_{ij}^t|\} \leq \epsilon$ Stop ;
Otnerwise $t = t + 1$ and return to Step 1.

We demonstrate the convergence of the algorithm to the unique primal optimal solution of the problem. We also show that the dual variables $\{(\alpha^t, \beta^t, \gamma^t, \delta^t, \psi^t)\}_{t=0}^{\infty}$ are bounded and converge as well to an optimal dual solution.

We consider next the multicommodity strictly convex entropy cost transportation problem which is formulated as follows:

$$\min \ [\sum_{i=1}^{n}\sum_{j=1}^{m}(\sum_{k=1}^{K} x_{ijk}\ln(\frac{x_{ijk}}{c_{ijk}}) + (U_{ij} - \sum_{k=1}^{K} x_{ijk})\ln(\frac{U_{ij} - \sum_{k=1}^{K} x_{ijk}}{U_{ij} - \sum_{k=1}^{K} c_{ijk}}))] \tag{8}$$

subject to (2),(3),(5) and (6) and for values of c_{ijk} which satisfy

$$0 < c_{ijk}, \forall(i,j,k), \tag{9}$$

and

$$\sum_{k=1}^{K} c_{ijk} < U_{ij}, \forall(i,j). \tag{10}$$

The first order optimality conditions of the corresponding Lagrangian dual of (8), (2),(3) are (since (5),(6) will be satisfied by the optimal solution):

$$\ln(\frac{x_{ijk}}{c_{ijk}}) - \ln(\frac{U_{ij} - \sum_{k=1}^{K} x_{ijk}}{U_{ij} - \sum_{k=1}^{K} c_{ijk}}) - \alpha_{ik} + \beta_{jk} = 0, \forall(i,j,k) \tag{11}$$

As a consequence, by letting $R_{ik} = \exp(\alpha_{ik}), \forall(i,k)$, and $S_{jk} = \exp(-\beta_{jk}), \forall(j,k)$, the optimal primal variables are given by the expression

$$x^{*}_{ijk} = \frac{U_{ij}}{U_{ij} - \sum_{p=1}^{K} c_{ijp} + \sum_{p=1}^{K} R_{ip}c_{ijp}S_{jp}} R_{ik}c_{ijk}S_{jk}, \forall(i,j,k) \tag{12}$$

The algorithm that we develop for finding the optimal solution of (8) s.t. (2),(3),(5) and (6), is a generalization of the well known RAS algorithm for matrix balancing. It is stated below:

Algorithm for Multicommodity Entropy Cost Transportation Problem

Step 0 (Initialization) :

$:= 1;\ R^t_{ik} = 1, \forall(i,k);\ S^t_{jk} = 1, \forall(j,k).$

Step 1 (Balancing rows) :

$$R^{t+1}_{ik} = \frac{O_{ik}}{\sum_{j=1}^{m}\left(\frac{U_{ij}}{U_{ij}-\sum_{p=1}^{K} c_{ijp}+\sum_{p=1}^{K} R^t_{ip} c_{ijp} S^t_{jp}}\right) c_{ijk} S^t_{jk}}, \forall(i,k)$$

Step 2 (Balancing columns) :

$$S^{t+1}_{jk} = \frac{D_{jk}}{\sum_{i=1}^{n}\left(\frac{U_{ij}}{U_{ij}-\sum_{p=1}^{K} c_{ijp}+\sum_{p=1}^{K} R^{t+1}_{ip} c_{ijp} S^t_{jp}}\right) c_{ijk} R^{t+1}_{jk}}, \forall(j,k)$$

Step 3 (Stopping criterion):

If $\max\{\max_{ik} \frac{|R^{t+1}_{ik}-R^t_{ik}|}{R^t_{ik}}, \max_{jk} \frac{|S^{t+1}_{jk}-S^t_{jk}|}{S^t_{jk}}\} \leq \epsilon$, go to step 4;
Otherwise $t = t + 1$ and return to Step 1.

Step 4 (Computation of the primal solution) :

x^*_{ij} is computed by (12); Stop.

We demonstrate the convergence of the last algorithm to the unique primal optimal solution of the multicommodity entropy problem. We note that the solution is an interior feasible point. We also show that the dual variables $(\alpha_t, \beta_t)_{t=0}^{\infty}$ are bounded and converge as well to an optimal dual solution.

Then, we imbed the above algorithms respectively in a proximal minimization approach and an entropy programming approach in order to solve multicommodity general convex cost and linear cost transportation problems. We extend this class of methods for solving multicommodity minimum cost problems on transhipment networks with upper bounds on the arc flows by using node splitting and arc reversal transformations to obtain equivalent multicommodity transportation problems.

Computational results are given for serial and coarse grain parallel implementations of these algorithms for both nonlinear and linear cost problems.

Management of Data Structures for Large-Scale Transportation Networks

N. Christofides H. O. Badra* Y. M. Sharaiha

Operational Research and Systems
The Management School
Imperial College of Science, Technology & Medicine
Exhibition Road, London SW7 2PG
United Kingdom

1. Abstract

There has been growing interest in computer graphics applications, which has made it necessary to develop efficient data structures and algorithmic procedures for handling large-scale graphical information in real-time. A number of tree data structures have been successfully applied to represent and manage image and graphics data. Among them, the k-d tree (Bentley [1975]), 2^N-division tree (Samet [1980] and Meagher [1982]), and BD-tree (Ohsawa and Sakauchi [1983]) are noteworthy from the view point of coping with a large amount of data in a dynamic environment. In particular, such representations are computationally efficient in terms of retrieval, processing and manipulation. It is often the case, particularly in the representation of large-scale networks, that the entire tree cannot be maintained in main memory, and thus paging from disk becomes necessary. A variety of schemes have been proposed (see *e.g.*, Shaffer *et al* [1993] and Henrich *et al* [1990]) to overcome this problem by paging from disk. In this paper, we first present a new dynamic structure for the representation of large-scale networks (F-tree) and then present a scheme for its memory management.

* Correspondence to: Hala Badra
The Management School, Imperial College, 53 Prince's Gate,
Exhibition Road, London SW7 2PG, UK
e-mail: h.badra@ic.ac.uk
Tel: +44 (0)71 589 5111 (ext. 7129)
Fax: +44 (0)71 823 8134

2. F-tree Data Structure

In Christofides, Badra & Sharaiha [1993], we developed two complementary data structures for the representation of networks: a *topological* and a *topographical* representation. Together, they support efficient graph-theoretic and geometric operations. A topological representation facilitates the implementation of graph-theoretic optimization algorithms under dynamic on-line editing operations. A new graph structure, termed the *Dynamic Forward Star (DFS)*, was developed and based on the well-known *(Static) Forward Star (SFS)* so the algorithms in the literature can be readily modified. A topographical representation is necessary to support efficient spatially-based geometric operations that require a different labelling scheme from that of the graph structure. We classify geometric operations into two categories: (i) Standard geometric operations of network *editing* which involve the identification of *"nearest"* arc (and/or vertex), and their *insertion (or deletion)*; and (ii) advanced geometric operations involving high-speed *range retrieval*, and *splitting* and *merging* of sub-networks.

The topographical data structure, termed the *Folded tree (F-tree)*, presented in Christofides *et al* [1993] is an extension of the *Binary Division tree (BD-tree)*, first introduced by Ohsawa & Sakauchi [1983]. The F-tree is used for the representation of networks by recursive partitioning of the space into rectangular regions. Figure 1 presents an example of an F-tree. Figure 1(a) shows the spatial decomposition of points (vertices or arc centroids of the network), and Figure 1(b) shows the tree structure representing the decomposition process. Each region is constructed dynamically by adhering to some predetermined rules regarding decomposition. The resulting *binary tree* structure is composed of *internal* and *leaf* nodes. The tree structure continues to grow as additional data is represented. A *zone identifier* is used to illustrate how the regions have been partitioned. This is a binary string generated by assigning (recursively) 0 if the region lies in the left half of the bisector parallel to the y axis or lower half of the bisector parallel to the x axis, and assigning 1 if it corresponds to the right or upper half. The binary sequence also represents the path from the root node to any other node, with 0 for the path to the left node and 1 for the path to the right node. Moreover, each internal node also has a *circumscribed quadrilateral* which includes all data located under the node (see, for example, nodes 5 and 11 in Figure 1(b)). In that paper, algorithms for the application of the above geometric operations on the F-tree were presented. The data structure was evaluated on the following functions: (i) The properties of the tree structure; (ii) the efficiency of the data structure in supporting standard geometric operations of insertion, deletion and nearest arc; and (iii) the efficiency of the data structure in performing advanced geometric operations of range retrieval.

The data structure was tested on a case study of the road map of Italy, with four network sizes (I), (II), (III), and (IV). Figure 4 illustrates the data-file content of the largest network size (case I), with an image size of $(2^{15}\times2^{15})$, and a graph size of $n=12603$ vertices and $m=20693$ arcs. The three network sizes (II, III, and IV) represent sections of the complete road network (I) reflecting smaller regions chosen in the northern part of Italy (36, 16 and 8% of the original network size). Table 1 includes the main characteristics of the constructed vertex and arc F-trees for all network sizes. Computational results for range retrieval on the road network (I) are presented in Section 4.

3. Memory-Managed Data Structure

For large-scale networks (eg size I), it is often the case that the entire F-tree cannot be strored in *main memory* and *paging* from *disk* becomes necessary. In this section, we propose a procedure for efficient memory management based on a systematic partitioning of the tree during its construction into smaller *subtrees*, each subtree representing a corresponding *subnetwork*. A *tree of subroots* is created dynamically which has the property of preserving the *neighbourhood* relationships of the original F-tree. This scheme becomes particulary suitable for search algorithms in which a minimum number of external pages must be traversed since paging from disk is computationally expensive. Figure 2 represents the tree of subroots for the F-tree of Figure 1(b). The tree of subroots is constructed as follows. A fixed *capacity* Q is chosen to be the maximum number of nodes in each subtree. A subtree is defined by the tree rooted at its subroot upto, but not including, any other subroot. A *weight* $w(i)$ is associated with each node, where $w(i)$ is the number of nodes in the subtree rooted at i, including i itself. Hence, the weight of any leaf node is equal to one. Next, we describe the management of the tree of subroots.

3.1 Subtree management

Let r be the subroot of a subtree $T_r \subset T$, where T is the entire F-tree. For any subroot r, $w(r) \leq Q$, where Q is user-defined such that it utilizes the maximum storage space occupied by a given subtree to be paged to and from disk. Furthermore, we aim to maintain the average weight of each subroot r to be approximately $\frac{Q}{2}$. We also assign a lower bound L on the size of each subtree. We choose $L = \frac{Q}{4}$ in order that we limit the number of subtrees that contain a small number of nodes. This is consistent with the requirement of maintaining a balance in the amount of data stored in each subtree and thus reduce the number of major adjustments on *overflow* and *underflow* of pages under dynamic updates to the tree.

Splitting of two subtrees

Let r be the subroot of the subtree T_r. If at any time, for a subroot r , $w(r) > Q$, then the subtree rooted at r is *split* into two subtrees, an *upper level* subtree T'_r and a *lower level* subtree T'_{r^*}. This may occur after an addition of a node, or after a deletion of a node leaving a subtree with very few nodes which may then be a candidate for merging with the tree above which, in turn, would create the upperbound bottleneck. The choice of r^*, the subroot of the lower level subtree, is dependent on the node weights. As noted above, this is chosen to be the node in the subtree to be split which has a weight closest to $\frac{Q}{2}$. This subroot is identified by starting with r and traversing down the upper level subtree along the branch containing the heavier node, until the split node (or new subroot) is met. Figure 3(a) illustrates how the two subtrees, T'_3 and T'_{17}, are split from T_3. The weight of all nodes on the path (highlighted in Figure 3(a)), between the two subroots (nodes 3 and 17) is updated.

Merging of two subtrees

Let r^* and r be two subroots such that r is the subroot of the subtree just above r^*. If at any time, $w(r^*) \leq L$ then we check if $w(r^*) + w(r) \leq Q$. If this is the case, then the subtrees T_{r^*} and T_r are *merged* into T'_r only. This may occur after deletion of a node from the subtree rooted at r^* which has very few nodes. However, if r^* is the main root of T then r does not exist and it is excluded from merging; thus no action takes place. On the other hand, the root may be involved in merging if it is r. The two subtrees, T_{r^*} and T_r , are also merged if $w(r^*) = 1$, regardless of whether T_r has reached its capacity. If the resulting tree has $w(r) > Q$, it is split using the splitting rules described before. Figure 3(b) illustrates how two subtrees, T_3 and T_{17}, are merged. The weight of all nodes on the path (highlighted in Figure 3(b)) between the two subroots (nodes 3 and 17) is updated by adding the weight of the subroot in the lower level subtree (node 17) to all node weights on that path.

Tree of subroots

As illustrated in Figure 2, the tree of subroots constructed using the rules above (splitting and merging) is generally not a binary tree. However, the *pointer system* of the tree of subroots preserves the original concept of the F-tree. Two pointers are required for each subroot to store the tree of subroots.

Subroot Son: The subroot son pointer of an internal node points to its first *left son* node in the tree of subroots. In the case of a leaf node, the subroot son pointer points to itself, flagged by a negative sign for distinction from the internal nodes.

Subroot Brother: The subroot brother pointer of a node points to its *right brother* node. If the node has no right brother, then the subroot brother pointer points to its *father* node, and is flagged by a negative sign for distinction. The subroot brother pointer of the root node is set to zero.

3.2 Cache management

In this section, we describe a caching system for maintaining data in main memory and addressing data on disk as required.

3.2.1 Data on disk

The entire F-tree is stored on disk in a *subtree-file* which is divided into a collection of *pages*. Each page contains a complete self-contained subtree indexed by its subroot node number in the tree of subroots, and contains all information on each of its nodes.

3.2.2 Data in main memory

The tree of subroots (*subroots-table*) and a subset of the total number of subtrees and their associated information (*nodes-table*) are both kept in main memory.

Nodes-table

The nodes-table is broken into a collection of *blocks*. Each block holds the latest information on a single subtree. The number of subtrees kept in main memory at any one time and therefore the size of the nodes-table can be chosen according to the type of machine used. In order to perform all editing operations in main memory prior to splitting of subtrees, we choose the capacity of each block to be $Q+2$. The nodes of a subtree are maintained in sorted order in each block so that a *binary search* can be used to locate a particular node within a block once it is addressed

Subroots-table

The tree of subroots is always kept in main memory in the subroots-table. For each subroot, the table consists of the following arrays:

(i) The *subroot* node number in the F-tree.

(ii) The *page* number in the subtree file, indicating the storage position of that subtree on disk.

(iii) The *block* number in the nodes-table, indicating the storage position of that subtree in main memory. If the subtree is not in main memory, this is set to zero.

(iv) The pointers *Subroot Son* and *Subroot Brother* required to define the structure of the tree of subroots as described above.

(v) The *ranking* of the subtree according to the "*least recently used*". In addition, a negative sign identifies whether the subtree has been *modified* since it was last read in to main memory.

3.2.3 Mode of Caching

Once a node is addressed and is not in main memory, the subtree which contains it is accessed from disk. The mode of *caching* we use is first based on the identification of whether the node in question is in main memory or on disk. We represent the node number using a 32-bit binary string. The first 19-bits of this binary string contain the actual node number and the remaining 13-bits contain the subroot node number to which the node belongs. Thus, given this representation, we can identify from the subroots-table whether the node is in main memory or on disk using the Block(.) array. If the node is not in main memory, the subtree on disk which contains it is cached. It then replaces a subtree in main memory based on the "least recently used". If the node is in main memory, a binary search is carried out on its node-table block to locate the node and its details.

4. Computational Results

In this section, we examine the properties of the subtrees and the tree of subroots created by applying the partitioning scheme described above. We also investigate the efficiency of this scheme in implementing range retrieval (or zooming) algorithms when caching from disk is necessary. This is the case for large zoom areas of the road map of Italy (Figure 4), which was used as a case study. The retrieval algorithm implemented was run on a PC 80486 co-processor.

In Table 2, the main properties of the resulting vertex and arc subtrees for varying values of Q are presented. The resulting average size (column (2) and (6)) is consistent with the aim of creating subtrees of average size approximately equal to $\frac{Q}{2}$. It is also guaranteed that each subtree has more than $\frac{Q}{4}$ nodes. This is an important factor since *ill-balanced* subtrees may cause inefficient memory utilization for node splitting and merging. Table 2 also provides an indication of the balance of each of the subtrees in terms of maximum (column (3) and (7)) and average (column (4) and (8)) *height*, where the height is defined by the branch cardinality of the path from the subroot to a leaf node. The experimental results support the claim of good balance using this partitioning scheme. The main properties of the resulting vertex and arc tree of subroots are also presented in Table 3 for varying values of Q. In all experimentation, we use a constant number of blocks (i.e., subtrees in main memory) to be fifty for all values of Q. The tree of subroots constructed is then used for testing the retrieval algorithm.

In testing the retrieval algorithm, we specify a rectangular *window* varying in size from 5 to 80%. For each window size, the number of arcs which overlap wholly or partly with that window is identified. We note that this wide window size variation allows for a balanced testing of this algorithm since the output number of arcs, given as a percentage of the total number of arcs, varies between less than 1% (for window size of 5%) upto 80% (for window size of 80%). Table 4 shows the real-time efficiency of the procedure. The Table also shows how the CPU time decreases rapidly as the maximum size of each subtree increases.

5. Conclusion

The primary motivation for our work was to effectively manage the topological storage of large-scale transportation networks requiring the use of paging to and from disk. This paper presented an overview of a new scheme for efficient memory management of the F-tree data structure by partitioning the entire tree in to smaller subtrees. We have described this partitioning scheme in relation to the F-tree and presented a memory-managed data structure for its implementation. However, with minor modifications, the scheme can be adapted to manage a variety of tree-based data structures. We showed how each subtree can be handled as a separate entity of the network each indexed by a subroot. Moreover, the tree of subroots preserves the topology and neighbourhood relationships of the original tree. The scheme was tested against road map data from Italy. Computational results presented include the charactersitics of the tree-based structures and confirm the real-time efficiency of the retrieval algorithm.

6. References

[1] Bentley, J. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM,* Vol. **18**, No. **9**, 509-517.

[2] Christofides, N., Badra, H.O. and Sharaiha, Y.M. (1993). Data structures for topological and geometric operations on large-scale networks. *Imperial College Technical Paper No SWP9326/OR*, submitted for publication.

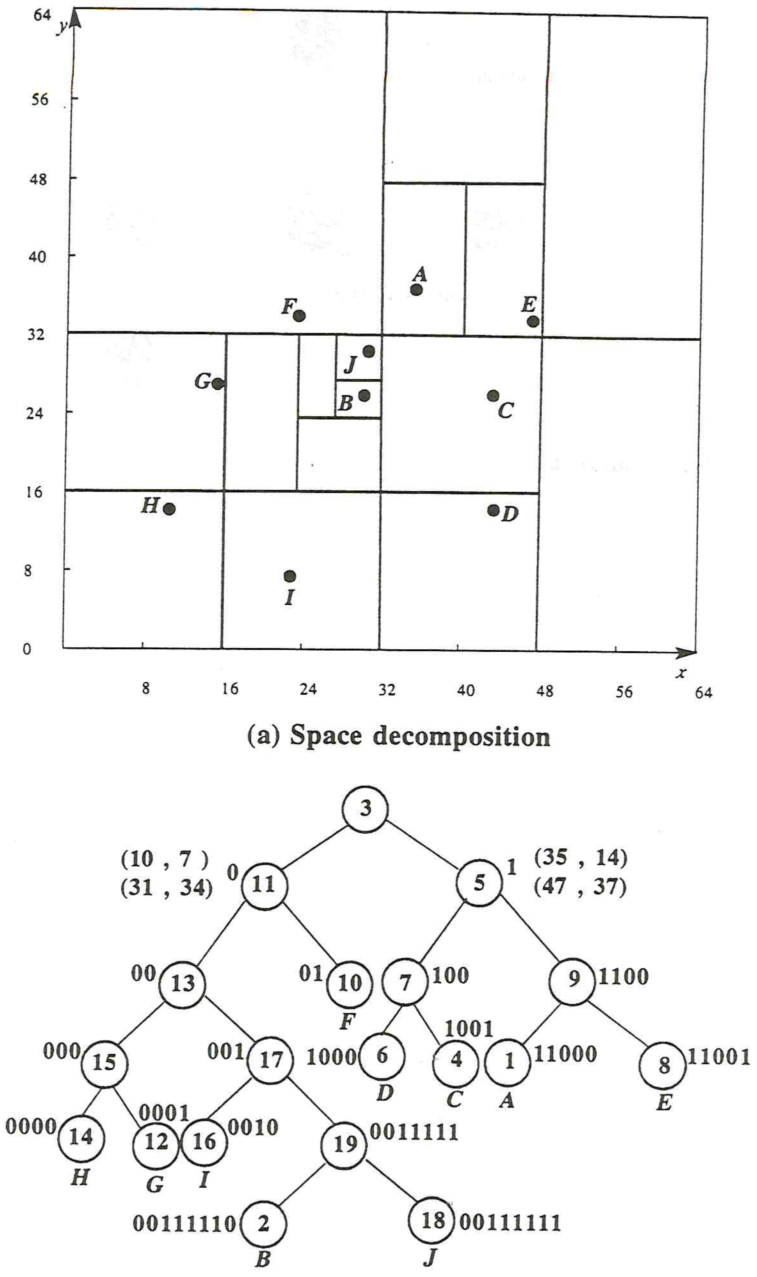
[3] Henrich, A., Six, H. and Widmayer, P. (1990). Paging binary-trees with external balancing. *Lecture Notes in Computer Science*, Vol. **411**, 260-276.

[4] Meagher, D. (1982). Geometric modelling using octree encoding. *Computer Graphics and Image Processing*, Vol. **19**, No. **2**, 129-147.

[5] Ohsawa, Y. and Sakauchi, M. (1983). The BD-tree - A new N-dimensional data structure with highly efficient dynamic characteristics. *The Proceedings of the IFIP 9th World Computer Congress*, 539-544.

[6] Samet, H. (1980). Region representation: quadtrees from boundary codes. *Communications of the ACM,* Vol. **23**, No. **3**, 163-170.

[7] Shaffer, C. and Brown, P. (1993). A paging scheme for pointer-based quadtrees. *Lecture Notes in Computer Science*, Vol. **692**, 89-104.



(a) Space decomposition

(b) Tree structure

Figure 1. Example of the F-tree structure

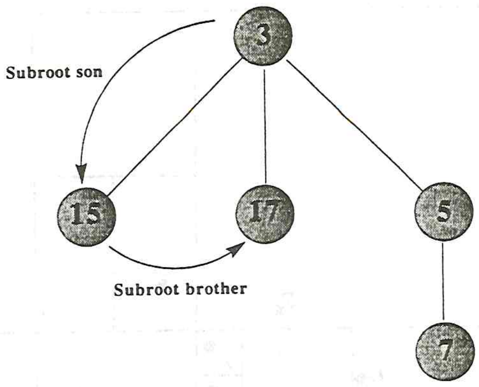
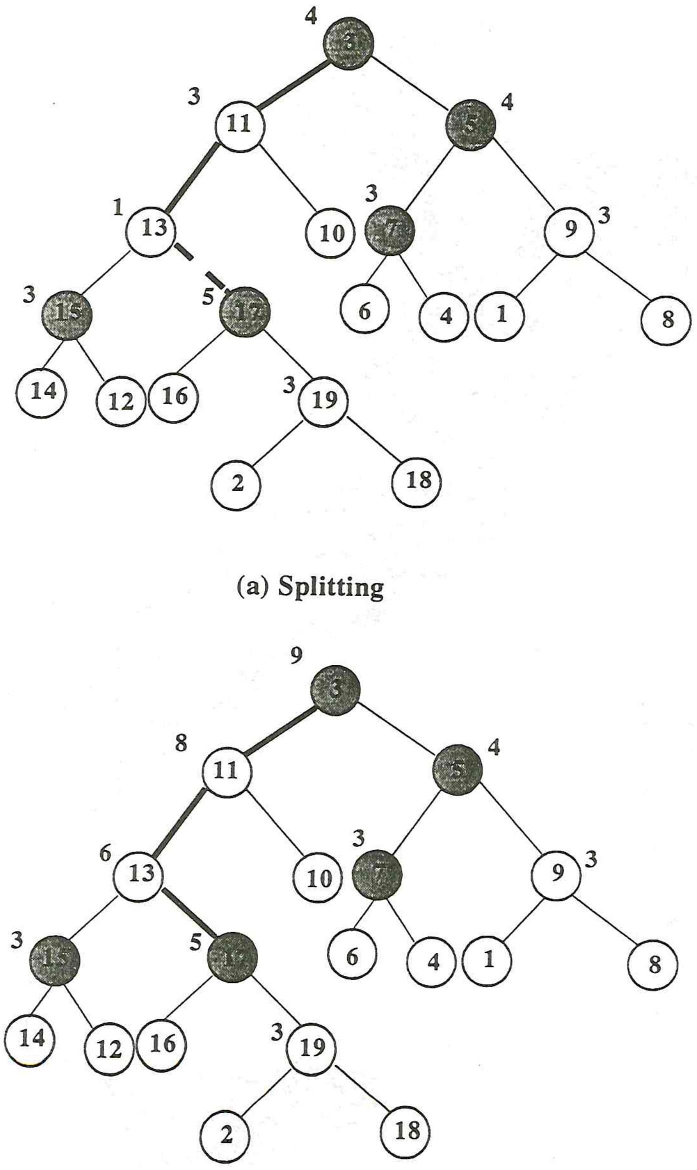


Figure 2. Tree of subroots



(a) Splitting

(b) Merging

Figure 3. Updating of weights after (a) splitting; (b) merging

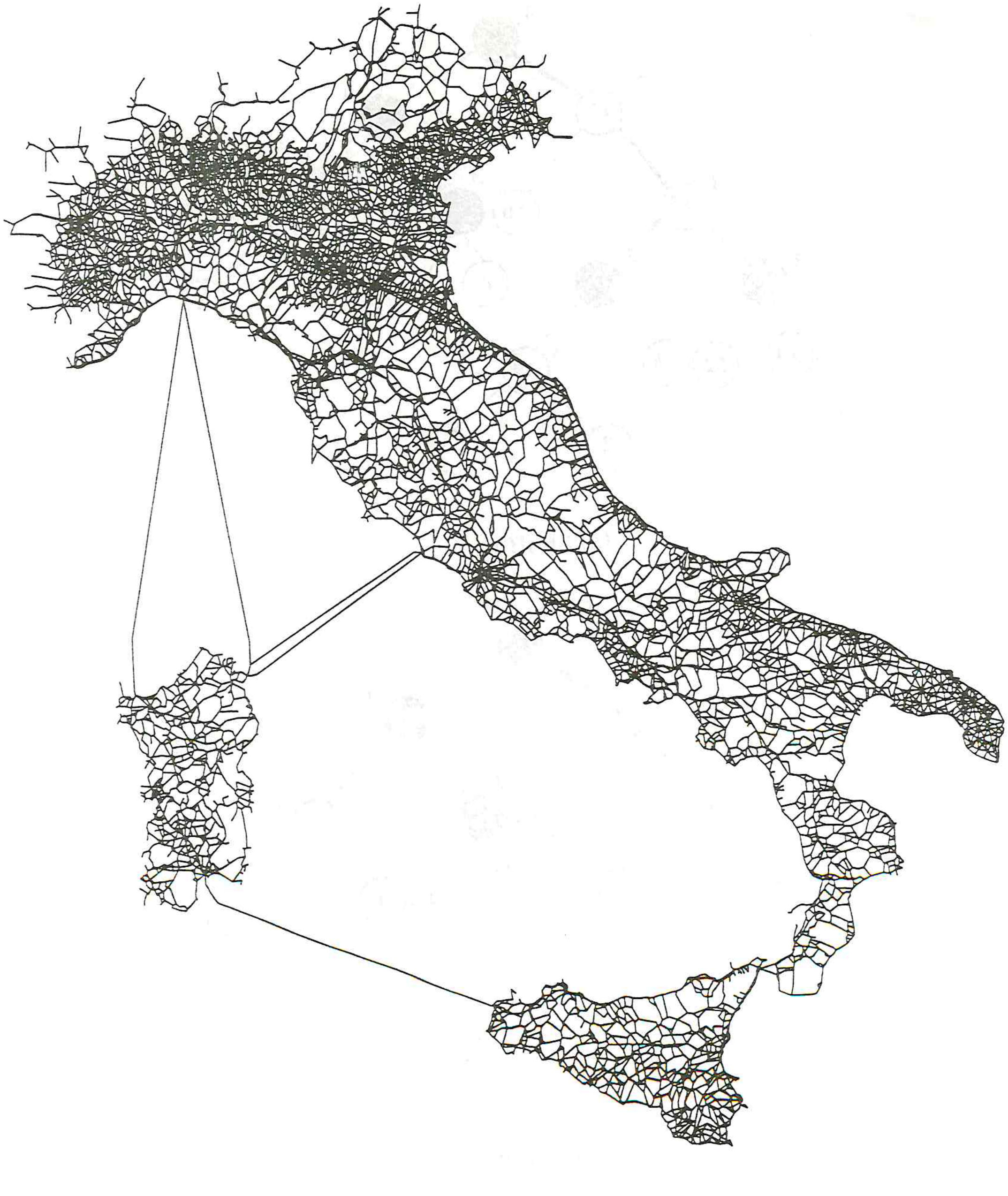


Figure 4. Road map of Italy - Network size I

Table 3. **Dimensions of the constructed vertex and arc tree of subroots**

Max Capacity Q	Vertex Tree of Subroots				Arc Tree of Subroots			
	(1) Max height	(2) Avg height	(3) Max width	(4) Avg width	(5) Max height	(6) Avg height	(7) Max width	(8) Avg width
20	13	5.75	471	164.62	11	6.20	896	325.09
60	9	4.63	158	79.78	12	5.14	277	99.92
100	8	4.30	122	48.11	9	4.11	208	79.44
400	6	3.57	34	18.67	7	3.33	58	24.71
600	8	3.40	26	9.88	7	3.28	41	17.29

Legend:

(1) and (5) : Maximum height of vertex and arc tree of subroots.
(2) and (6) : Average height of vertex and arc tree of subroots.
(3) and (7) : Maximum width of vertex and arc tree of subroots.
(4) and (8) : Average width of vertex and arc tree of subroots.

Table 4. **Computational time (CPU sec) for range retrieval**

Max Capacity Q	Window Size (as a percentage of the image size)				
	5%	10%	20%	40%	80%
20	0.22	0.33	0.66	3.46	24.78
60	0.22	0.22	0.33	1.65	11.32
100	0.22	0.22	0.39	1.37	8.68
400	0.22	1.65	0.22	1.15	5.28

Table 1. Dimensions of the constructed vertex and arc F-trees

Network Size	Vertex F-tree			Arc F-tree		
	(1) # Nodes	(2) Max height	(3) Avg height	(4) # Nodes	(5) Max height	(6) Avg height
I (100%)	25205	28	16.52	41385	29	17.42
II (36%)	9075	25	14.88	15069	26	15.80
III (16%)	4005	18	13.87	6827	20	14.90
IV (8%)	2125	16	12.33	3509	18	13.38

Legend:

(1) and (4) : Number of nodes in the vertex and arc F-trees.

(2) and (5) : Maximum height (in number of branches in the path from the root to the leaf nodes) of the vertex and arc F-trees.

(3) and (6) : Average height (in number of branches in the path from the root to the leaf nodes) of the vertex and arc F-trees.

Table 2. Dimensions of the constructed vertex and arc subtrees

Max Capacity Q	Vertex Subtrees				Arc Subtrees			
	(1) Max size	(2) Avg size	(3) Max height	(4) Avg height	(5) Max size	(6) Avg size	(7) Max height	(8) Avg height
20	19	11.77	14	5.89	19	11.57	13	5.78
60	59	35.06	31	17.53	59	34.49	33	17.24
100	99	58.34	51	29.17	99	57.80	56	28.90
400	389	223.05	197	111.52	399	237.84	200	118.92
600	583	315.06	292	157.53	599	339.22	300	169.61

Legend:

(1) and (5) : Maximum size of the vertex and arc subtrees.

(2) and (6) : Average size of the vertex and subtrees.

(3) and (7) : Maximum height of the vertex and arc subtrees.

(4) and (8) : Average height of the vertex and arc subtrees.

Finito di stampare giugno 1994 - Tipografia ESAGRAFICA s.r.l.
(00184) Roma - Via della Polveriera, 12/13/13a - Tel. 4885197-4882678