A Scalable Logic-based Benders Decomposition to Optimize a Dynamic Demand-Responsive Transport System

L. Zigrand[a,b,*], R. Wolfler Calvo[b,c], E. Traversi[b] and P. Alizadeh[d]

[a] Padam Mobility, Paris, France
louis@padam.io
[b] LIPN (CNRS – UMR 7030), Université Sorbonne Paris Nord, Paris, France
zigrand@lipn.univ-paris13.fr, wolfler@lipn.univ-paris13.fr, traversi@lipn.univ-paris13.fr
[c] DIM, Università di Cagliari, Cagliari, Italy
roberto.wolflerc@unica.it
[d] Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France
pegah.alizadeh@devinci.fr
* Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 2, 2022

Keywords: Demand-Responsive Transport, Logic-based Benders Decomposition

1 Introduction

Demand Responsive Transport (DRT) defines shared transport systems where the vehicles adapt their routes dynamically to the demand rather than using fixed routes and timetables. The motivation of using such hybrid models is twofold: they straddle conventional public transport and taxis, as their schedules and routes are quite flexible, and they constitute a viable alternative to individual transport due to a lower cost of use.

Most of the time, such systems allow their users to make transport requests days before, as well as during the day of the service. Consequently, if we put ourselves at the start of the service and take a look at the demand, we can see two groups: the past ones that have already been made, namely *offline* requests, and the future ones that have still to be made, namely *online* requests. From there, a natural question can arise: is it possible to optimize the dispatching of the offline requests while taking into account the online requests if we know them beforehand? If so and to go a little bit further, is the gain in pooling of the service's vehicles substantial in comparison to the usual planning optimization using myopic objective functions?

This work has been performed jointly with Padam Mobility, a well-established company that provides technological support to DRT companies. We first propose a generalization of the Dial-a-Ride Problem (DaRP) model to describe the previously presented situation. We then outline a Logic-based Benders Decomposition method to solve our problem and show that this approach is efficient and adapted to the reality of an actual DRT service's structure. Finally, we evaluate the performance difference between a service where the offline requests have been organized with regards to a myopic objective function versus with regards to the online requests.

2 Problem Description

We consider a service where each agent $k \in \mathcal{K}$ has a vehicle with a capacity $Q_k \in \mathbb{N}^\star$, starts to work at $W_k^+ \in \mathbb{R}_+$ from depot D_k^+ and finishes their shift at $W_k^- \in \mathbb{R}_+$ in depot D_k^-. Regarding the travel requests $\mathcal{R}$ associated to this service, we split them between the offline $\mathcal{R}_{\text{Off}}$ and the online $\mathcal{R}_{\text{On}}$ ones. In both cases, we describe a request $r \in \mathcal{R}$ with a pick-up P_r and a drop-off D_r virtual nodes and we associate to each $n \in \{P_r; D_r\}$ a time-window of service $[E_n; L_n] \subset \mathbb{R}_+$, a load variation Q_n as well as a service time $S_n \in \mathbb{R}_+$. Finally, each $r \in \mathcal{R}$ is given a maximum ride time $M_r \in \mathbb{R}_+$ and each $r \in \mathcal{R}_{\text{On}}$ has a probability of actually happening $\mathbf{P}(r)$.

Regarding the underlying graph of this Vehicle Routing Problem, we note $\mathcal{N}$ the set of all the virtual nodes previously described and $\mathcal{N}_=$ the pairs of virtual nodes associated to the same physical node. Lastly, we note $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$ the set of edges that can be crossed during the day and, for each $(n, n') \in \mathcal{A}$, we define $\Delta t_{n,n'}$ the expected travel time between both nodes.

Based on this construction, we can consider the following decision variables of our problem:

- $x_{k,i,j}$: Binary variable equal to 1 if and only if agent $k \in \mathcal{K}$ is expected to cross $(i, j) \in \mathcal{A}$.
- $q_{k,i}$: Continuous variable equal to the load of agent $k \in \mathcal{K}$ when departing from $i \in \mathcal{N}$.
- $s_{k,i}$: Continuous variable equal to the time when agent $k \in \mathcal{K}$ arrives at $i \in \mathcal{N}$.

Based on these definitions as well as the traditional three-index formulation (Cordeau, 2006), we model our problem with the following Mixed-Integer Linear Programming formulation:

$$\max_{x,s,q} \sum_{r \in \mathcal{R}_{\text{On}}} \sum_{k \in \mathcal{K}} \sum_{\{j \in \mathcal{N} | (P_r; j) \in \mathcal{A}\}} x_{k,P_r,j} \times \mathbf{P}(r), \text{ subject to:} \tag{1}$$

$$\sum_{k \in \mathcal{K}} \sum_{\{j \in \mathcal{N} | (i;j) \in \mathcal{A}\}} x_{k,i,j} = 1 \qquad \forall i \in P_{\text{Off}} \cup D_{\text{Off}} \tag{2}$$

$$\sum_{k \in \mathcal{K}} \sum_{\{j \in \mathcal{N} | (i;j) \in \mathcal{A}\}} x_{k,i,j} \leq 1 \qquad \forall i \in P_{\text{On}} \cup D_{\text{On}} \tag{3}$$

$$\sum_{\{j \in \mathcal{N} | (P_r;j) \in \mathcal{A}\}} x_{k,P_r,j} - \sum_{\{j \in \mathcal{N} | (D_r;j) \in \mathcal{A}\}} x_{k,D_r,j} = 0 \qquad \forall k \in \mathcal{K},\ \forall r \in \mathcal{R} \tag{4}$$

$$\sum_{\{j \in \mathcal{N} | (D_k^+;j) \in \mathcal{A}\}} x_{k,D_k^+,j} = 1 \qquad \forall k \in \mathcal{K} \tag{5}$$

$$\sum_{\{i \in \mathcal{N} | (i;D_k^-) \in \mathcal{A}\}} x_{k,i,D_k^-} = 1 \qquad \forall k \in \mathcal{K} \tag{6}$$

$$\sum_{\{j \in \mathcal{N} | (j;i) \in \mathcal{A}\}} x_{k,j,i} - \sum_{\{j \in \mathcal{N} | (i;j) \in \mathcal{A}\}} x_{k,i,j} = 0 \qquad \forall k \in \mathcal{K},\ \forall i \in P \cup D \tag{7}$$

$$\begin{aligned} &s_{k,i} + S_i + \Delta t_{i,j} - s_{k,j} - (1 - x_{k,i,j}) \times M_{i,j} \leq 0, \\ &\text{where } M_{i,j} = \max\{0; L_i + S_i + \Delta t_{i,j} - E_j\} \end{aligned} \qquad \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{A} \setminus \mathcal{N}_= \tag{8}$$

$$s_{k,i} - s_{k,j} - (1 - x_{k,i,j}) \times \max\{0; L_i - E_j\} \leq 0 \qquad \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{A} \cap \mathcal{N}_= \tag{9}$$

$$s_{k,P_r} - s_{k,D_r} \leq 0 \qquad \forall k \in \mathcal{K},\ \forall r \in \mathcal{R} \tag{10}$$

$$s_{k,D_r} - s_{k,P_r} - S_{P_r} \leq M_r \qquad \forall k \in \mathcal{K},\ \forall r \in \mathcal{R} \tag{11}$$

$$E_i \leq s_{k,i} \leq L_i \qquad \forall k \in \mathcal{K},\ \forall i \in \mathcal{N} \tag{12}$$

$$W_k^+ \leq s_{k,D_k^+} \leq s_{k,D_k^-} \leq W_k^- \qquad \forall k \in \mathcal{K} \tag{13}$$

$$q_{k,i} + Q_j - q_{k,j} - (1 - x_{k,i,j}) \times \min\{Q_k; Q_k + Q_i\} \leq 0 \qquad \forall k \in \mathcal{K},\ \forall (i,j) \in \mathcal{A} \tag{14}$$

$$\max\{0; Q_i\} \leq q_{k,i} \leq \min\{Q_k; Q_k + Q_i\} \qquad \forall k \in \mathcal{K},\ \forall i \in \mathcal{N} \tag{15}$$

The objective presented in Equation (1) aims at maximizing the insertion rate of the online requests. Then, Equations (2) to (4) define the assignment constraints of requests to agents. The routing constraints are expressed by Equations (5) to (7) while the timing constraints are described by Equations (8) to (13) and the load constraints are in Equations (14) and (15). In particular, Equation (9) manages the dwell time computation with regards to the pairs of virtual nodes that share the same physical location, which is a particularity of actual DRT services.

3 Methodology

The problem described in Section 2 is actually complex to solve as it is. In the literature, various approaches to solve the traditional DaRP model through the use of exact methods have been proposed to tackle this scaling issue: Branch-and-Cut, Branch-and-Price and Branch-and-Price-and-Cut paradigms notably (Ho *et al.*, 2018).

However, another way to look at this model is to decompose it as one assignment problem per agent. This is the reason why we thought of applying a Logic-based Benders Decomposition (Hooker & Ottosson, 2003) approach to this problem. Furthermore, recent studies using this technique on similar problems have shown promising results (Riedler & Raidl, 2018).

We present in Algorithm 1 our optimization framework based on Benders Combinatorial Cuts. The Master Problem assigns to each agent a set of requests to serve based on the already known assignment constraints. On their side, the Sub-Problems check those assignments, give a feedback to the Master Problem and try to repair heuristically the current assignment if necessary.

Algorithm 1: Logic-based Benders Decomposition Algorithm

```
Input:  - Agents $\mathcal{K}$
        - Offline and online requests $\mathcal{R}$
Output: Best feasible assignment $a \in \mathcal{P}(\mathcal{R})^{\mathcal{K}}$ found during the optimization process
1  function LogicBasedBendersDecomposition($\mathcal{K}$, $\mathcal{R}$)
2      master_problem ← MasterProblem($\mathcal{K}$, $\mathcal{R}$)
3      $a_{\text{Best}} \leftarrow \emptyset^{\mathcal{K}}$
4      while OptimalityIsNotProven(master_problem, $a_{\text{Best}}$) do
5          $(a_k)_{k\in\mathcal{K}}$ ← GetNewAssignment(master_problem)
6          for each $k \in \mathcal{K}$ do
7              $a_{k,\text{Off}}$ ← ComputeBCC(master_problem, $k$, $(a_k \cap \mathcal{R}_{\text{Off}})$)
8          $\left(a'_{k,\text{Off}}\right)_{k\in\mathcal{K}}$ ← RepairAssignments$\left((a_{k,\text{Off}})_{k\in\mathcal{K}},\ \mathcal{R}_{\text{Off}} \setminus \left[\bigcup_{k\in\mathcal{K}} a_{k,\text{Off}}\right]\right)$
9          for each $k \in \mathcal{K}$ do
10             $a'_k$ ← ComputeBCC$\left(\text{master\_problem},\ k,\ \left(a'_{k,\text{Off}} \cup [a_k \cap \mathcal{R}_{\text{On}}]\right)\right)$
11         if $\bigcup_{k\in\mathcal{K}} a'_{k,\text{Off}} = \mathcal{R}_{\text{Off}}$ then
12             $a''$ ← RepairAssignments$\left(\mathcal{K},\ a',\ \mathcal{R} \setminus \left[\bigcup_{k\in\mathcal{K}} a'_k\right]\right)$
13             if ObjectiveFunction($a''$) > ObjectiveFunction($a_{\text{Best}}$) then
14                 $a_{\text{Best}} \leftarrow a''$
15     return $a_{\text{Best}}$
```

In a few words, `MasterProblem` (l.2) initializes the Master Problem and builds heuristically a set of initial constraints, `ComputeBCC` (l.7 and l.10) computes new Benders Combinatorial Cuts and restricts the provided assignment to a feasible one if necessary, while `RepairAssignments` (l.8 and l.12) tries to insert the currently unserved requests into the current global assignment.

The initialization of the Master Problem plays a major role with regards to the convergence of the optimization process. One of our core focus has thus been to design an efficient heuristic in order to find Irreducible Infeasible Subsystems of requests to add to the initial pool of constraints. We also worked on improving the existing heuristics used to compute new Benders Combinatorial Cuts, to repair the broken assignments as well as to preprocess the underlying graph.

4 Preliminary Results

We tested our optimization framework based on Logic-based Benders Decomposition on instances from the literature (Cordeau, 2006, Riedler & Raidl, 2018) as well as on 25 real-world instances provided by our industrial partnership. On the latter, ranging from small instances of 4 agents, 30 offline and 30 online requests up to 30 agents, 100 offline and 200 online requests, our solver has been able to solve more than 80% of them to global optimality and obtain a gap of only a few requests on most of the other instances in less than an hour.

In another experiment, we considered for each instance two ways to optimize the planning of the agents at the start of the service's day: one where it is computed solely with the in-advance requests and the traditional objective function that minimizes the Total Duration of the Rides, and another one where it is designed by maximizing the insertion rate of the expected online requests using our solver. We then compared those initial schedules by simulating the arrival of the online requests during the service's day using a classical greedy online insertion module based on the Total Duration of the Rides minimization to answer the users. Our first results show that the initial plannings computed with our non-myopic objective function insert on average 11% more online requests than the ones computed with the traditional objective function. Furthermore, while the initial routes of the agents are on average 17% longer with our method, the final routes of the agents are only 7% longer on average. Those results show the potential of working on new non-myopic optimization techniques.

5 Conclusion

In this project, we first present a new Dial-a-Ride Problem-related model that describes more accurately real Demand Responsive Transport services than the existing models of the literature. Then, we propose a new Optimization Framework based on Logic-based Benders Decomposition to solve this problem and show that it scales well with actual instances, thanks to the structure of real services and newly designed heuristics. We also show encouraging results with regards to using such a paradigm to optimize the routes of the agents at the start of their day.

6 Acknowledgements

We thank Anh-Dung Nguyen from Padam Mobility for his precious support through this project.

References

Cordeau, Jean-François. 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, **54**(3), 573–586.

Ho, Sin C, Szeto, Wai Yuen, Kuo, Yong-Hong, Leung, Janny MY, Petering, Matthew, & Tou, Terence WH. 2018. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, **111**, 395–421.

Hooker, John N, & Ottosson, Greger. 2003. Logic-based Benders decomposition. *Mathematical Programming*, **96**(1), 33–60.

Riedler, Martin, & Raidl, Günther. 2018. Solving a selective dial-a-ride problem with logic-based Benders decomposition. *Computers & Operations Research*, **96**, 30–54.