Efficient Algorithms for the Multi-Period Line Planning Problem in Public Transportation

Güvenç Şahin[1]
Industrial Engineering,Sabanci University, Istanbul, Turkey
Zuse Institut Berlin, Germany

Amin Ahmadi Digehsara
Industrial Engineering, Sabanci University, Istanbul, Turkey

Ralf Borndörfer
Zuse Institut Berlin, Germany

Abstract

In order to plan and schedule a demand-responsive public transportation system, both temporal and spatial changes in demand should be taken into account even at the line planning stage. We study the multi-period line planning problem with integrated decisions regarding dynamic allocation of vehicles among the lines. Given the NP-hard nature of the line planning problem, the multi-period version is clearly difficult to solve for large public transit networks even with advanced solvers. It becomes necessary to develop algorithms that are capable of solving even the very-large instances in reasonable time. For instances which belong to real public transit networks, we present results of a heuristic local branching algorithm and an exact approach based on constraint propagation.

2012 ACM Subject Classification Applied computing → Transportation

Keywords and phrases public transportation, line planning, multi-period planning, local branching, constraint propagation

Digital Object Identifier 10.4230/OASIcs.ATMOS.2021.17

Category Short Paper

Funding *Güvenç Şahin*: Fellowship for Experienced Researchers from the Alexander von Humboldt Foundation.

1 Introduction

Responsive and flexible public transportation services become more indispensable as private services are under the radar for their detrimental effect on the environment. Earlier research on public transportation planning focused on fundamental issues such as identifying and framing the problems and constructing accurate models for these problems while integration of various problems associated with different planning stages has come forward [7] more recently along with advancement in research and computational power. While on-demand services, acclaimed for their responsiveness and utmost flexibility, are considered as the potential future of public transportation, it is accepted that they cannot replace the traditional public transit services. Yet, responsiveness of public services could be improved without sacrificing efficiency and effectiveness. In this respect, transit demand as the main driver should be pivotal in developing the plans and constructing the schedules for these services.

Traditional planning approaches consider demand as a static component particularly at higher levels of decision making. This is natural and plausible for the network planning and development stage. On the other hand, concurrent spatial and temporal changes in the

[1] Corresponding author.

21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021).
Editors: Matthias Müller-Hannemann and Federico Perea; Article No. 17; pp. 17:1–17:6
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

demand are critical for operational schedules. Recently, [6] propose a novel multi-period approach for the line planning stage supposing that line plans are the main link between strategic and operational plans and should be made with careful consideration of operational issues at the strategic level. With an effort to improve the demand-responsiveness of line plans, their primary contribution is the development of a multi-period line planning model which considers the changes in transit demand over time. Concurrently, they integrate tactical resource allocation constraints in order to ensure feasibility of multi-period line plans and exemplify this integration with dynamic allocation and assignment of vehicles to lines.

In their cost oriented multi-period approach with fixed costs of line selection and variable costs of service frequency on lines, the planning horizon is divided into discrete time periods each of which is associated with a different demand pattern. While the level of service on each line is determined for each period with the corresponding demand pattern, the periods are not independent from each other as they are coupled through the line selection decisions [6]. When compared to traditional static counterparts, allocation and assignment of resources to activities throughout the planning horizon are crucial in the case of multi-period planning [8]. As the activity levels (line frequencies) change from one period to the next, the resources (vehicles) are to be reallocated or reassigned. For vehicle scheduling and assignment, this can be achieved by discretizing the planning horizon as in [1] and [4]. Accordingly, a vehicle service (or a trip) is completed in one period; it can then be used on the same line or transferred to another. In the case of the latter, consideration of appropriate transfer time (i.e. the time it takes for the vehicle to travel from the ending station of one line to the starting station of another) is necessary.

In [6], it is shown that a multi-period approach is necessary when demand variation in time is a significant issue and also superior to a traditional approach that would combine line planning solutions of independent individual periods. However, computational challenges persist even at a higher level in comparison to single-period static line planning problems not only because of the convoluted structure of the multi-period line planning problem but also due to integration of vehicle transfer constraints. Out of the three PTN examples, finding optimal solutions for the largest one, namely the Quito Trolebus system, is not possible with a commercial solver. In this work, we discuss possible approaches that can be scaled to solve multi-period line planning problems with vehicle transfers even for a very-large PTN.

2 Problem Setting

An instance of the multi-period line planning problem presented in [6] is denoted by a public transportation network $PTN = (S, E)$ defined by a set of stations S and set of edges E connecting the stations, a set of time intervals T representing the planning horizon, transit demand d_e^t over the edges $e \in E$ in each period $t \in T$, and a set of potential lines L. A line $l \in L$ can be described as a path with a starting station and an ending station along with a subset of the edges to represent the path. Given the length of a discrete time period along with the starting and ending stations of lines, the transfer time from line l to line k is denoted by ρ_{lk} which should be calculated in multiples of time periods. In order to account for idle vehicles during a time period, an artificial line l_0 is used to represent a depot while $L_0 = L \cup \{l_0\}$.

The mathematical model in the form of a mixed integer programming problem formulation for the multi-period line planning problem with vehicle transfers (MPLPP-VT) includes a binary variable $y_l \in \{0, 1\}$ that takes value 1 if line l is selected, a non-negative integer variable v_l^t denoting the service level (and also corresponding to the number of vehicles dispatched) on line l in period t, and w_{lk}^{st} denoting the number of vehicles from line l used in period s to line k to be used in period t. Given that c_l^f and c_l^o denote respectively the fixed cost of selecting a line charged for the complete planning horizon and the operational cost on a line for each service in a period, the resulting formulation becomes

$$\min \quad \sum_{l \in L} c_l^f y_l + \sum_{l \in L} \sum_{t \in t} c_l^o v_l^t \tag{1}$$

$$\text{s.t.} \quad \sum_{l \in L_e} \mathcal{K} v_l^t \geq d_e^t \qquad \forall e \in E, \forall t \in T \tag{2}$$

$$\mathcal{W} y_l - v_l^t \geq 0 \qquad \forall l \in L, \forall t \in T \tag{3}$$

$$\sum_{\substack{k \in L_0 \\ t-\rho_{kl} \geq 0}} w_{kl}^{t-\rho_{kl},t} = v_l^t \qquad \forall l \in L, \forall t \in T \tag{4}$$

$$\sum_{\substack{k \in L_0 \\ t-\rho_{kl} \geq 0}} w_{kl}^{t-\rho_{kl},t} - \sum_{\substack{k \in L_0 \\ t+\rho_{kl} \leq |T|+1}} w_{lk}^{t,t+\rho_{lk}} = 0 \qquad \forall l \in L_0, \forall t \in T \tag{5}$$

$$\sum_{l \in L_0} \sum_{t \in T} w_{l_0 l}^{0t} = \mathcal{U} \tag{6}$$

$$\sum_{l \in L_0} \sum_{t \in T} w_{l l_0}^{t,|T|+1} = \mathcal{U} \tag{7}$$

$$y_l \in \{0,1\} \qquad \forall l \in L \tag{8}$$

$$v_l^t \in N \qquad \forall l \in L, \forall t \in T \tag{9}$$

$$w_{lk}^{st} \in N \qquad \forall l,k \in L_0, \forall s \in \{0\} \cup T, \forall t \in T \cup \{|T|+1\}, s < t. \tag{10}$$

The objective function (1) is to minimize the sum of total fixed costs for selecting lines and variable costs for providing service. Constraints (2) ensure that the demand on an edge in a period is covered by sufficient number of services with $\mathcal{K}$ denoting the capacity of a vehicle and L_e denoting the lines containing edge e in their path. Constraints (3) associate the line selections with service level decisions and put an upper bound $\mathcal{W}$ on the service level of a line in a period. Constraints (4) provide required number of vehicles to a line in each period considering all transfers including the vehicles that are already on the line (self-transfer represented with $w_{ll}^{t-1,t}$) and are to be retrieved from the depot. In each period, constraints (5) balance the vehicles transferred to and transferred from the line, again including self-transfers. Fleet size is controlled by constraints (6) and (7) ensuring that $\mathcal{U}$ vehicles are released from the depot at the beginning of the planning horizon, period 0, and all $\mathcal{U}$ vehicles are transferred back to the depot at the end of the planning horizon, period $|T|+1$.

Line planning problem is known to be NP-Hard, even for many special cases as shown in [9]. Therefore, computational challenges are expected to increase when many line planning problems are coupled with each other along with the addition of resource related constraints as exemplified for a very large instance of a real PTN in [6] which cannot be solved to optimality in reasonable time. Hence, it should be worthwhile to work on both heuristic and efficient exact algorithms.

3 Algorithms

Our earlier attempts focused on methods that rely on Benders' decomposition and Lagrangean relaxation. However, both methods failed to produce reliable algorithms. As a heuristic which still relies on solving the problem formulation (1)–(10), we use a local branching algorithm in its traditional form. As an exact solution approach, we propose an algorithm that solves a part of the problem formulation and adds missing constraints iteratively when they are violated only.

3.1 Local Branching

Local branching is an iterative method which may provide a high-quality incumbent solution within an acceptable computational time [3]. At each iteration, the original problem is divided into two sufficiently smaller sub-problems by generating so-called local branching cuts. The sub-problems include the feasible solutions of the original problem satisfying the additional local branching cuts. The algorithm may either identify a better feasible solution by solving the sub-problems within a short time or change the search region by a diversification mechanism. The algorithm terminates when some stopping criteria, i.e., the total time limit or the maximum number of diversifications, are reached. In the case of the MPLPP-VT, binary decision variables for line selection are used to partition the original solution space.

3.2 Logic-based Decomposition with Constraint Propagation

The spirit of our exact solution approach dates back to the original ideas in [2] for the TSP in the sense that we first eliminate a subset of the constraints, find a feasible solution with respect to the remaining constraints and identify which of the relaxed constraints are violated by this solution, and add the violated constraints to the problem formulation.

The algorithm iteratively continues in this fashion until no constraint violation is detected at an iteration. A critical feature of our algorithm is to explore only integer feasible solutions; hence, the integrality constraints are not relaxed. This idea of generating integer solutions for a relaxation of an integer programming problem formulation has been explored several times, particularly for the TSP. However, a straightforward implementation of such a scheme has only been presented recently in [5].

We adapt this idea to the MPLPP-VT and refer to this algorithm as logic-based decomposition with constraint propagation (LbDwCP). When constraints (5) are eliminated, the remaining problem is called the line planning subproblem (LPsP). The LPsP is solved to optimality. Given the line selection and service level decisions from the optimal LPsP solution, we check if the eliminated constraints associated with transfer of vehicles are satisfied; it is called the vehicle transfer feasibility problem (VTfP). If all constraints are satisfied, the solution of LPsP is also optimal for MPLPP-VT; otherwise, LPsP is extended with the selected violated constraints of VTfP and resolved to optimality. Figure 1 illustrates the mechanics of the algorithm.

4 Computational Results

We use the real PTN data from [6]; it includes the Istanbul Metrobus as the smaller problem with 44 stations and 9 lines (with three variants of the demand data), the Athens Metro as the medium-size problem with 51 stations and 59 lines and the Quito Trolebus as the

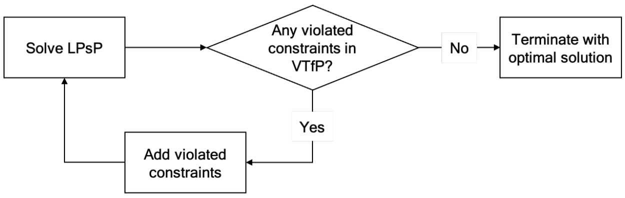

Figure 1 Average utilization and distribution of inbound truck sizes.

large-scale problem with 278 stations and 318 lines (Quito-318). We also generate a smaller version of the Quito Trolebus problem with the same network but only 122 lines (Quito-122) by eliminating some of the lines whose paths are already included in longer lines. We compare the performance of four alternative solution methods with the following settings:

- The commercial solver Gurobi is run on its default integer programming solver settings with a CPU time limit of 86400 seconds (1 day) for Quito-318.
- The local branching algorithm is limited with 20, 420, 1800 and 3600 seconds to solve a node problem and 60, 3600, 18000, and 86400 seconds for the total CPU time.
- Although the LbDwCP algorithm is designed to solve the LPsP problem to optimality in each iteration, the optimal LPsP solution cannot be found for the Quito-318 instance even at the first iteration. Therefore, a time limit of 3600 seconds is set to solve the LPsP in each iteration, and the best feasible solution found within this limit is used to check for the violated constraints.
- We also use Gurobi's built-in lazy constraints functionality as a benchmark approach for the LbDwCP algorithm; the violated constraints for every new integer incumbent solution are added within the branch-and-bound procedure employing the callback function. A CPU time limit of 86400 seconds (1 day) is set for Quito-318.

The results are shown in Table 1 where the Cost column shows the best feasible solution found while the Time columns shows the CPU time in seconds.

Table 1 Performance of alternative solution approaches.

	Gurobi		Local branching		LbDwCP		Lazy constraints	
Instance	Cost	Time	Cost	Time	Cost	Time	Cost	Time
Istanbul-1	61334.60	<1	61334.60	2	61334.60	6	61334.60	<1
Istanbul-2	48807.00	<1	48807.00	<1	48807.00	5	48807.00	1
Istanbul-3	35377.80	<1	35377.80	<1	35377.80	6	35377.80	1
Athens	68030.58	2070	68030.58	1050	68030.58	417	68030.58	286
Quito-122	21690.17	59134	21691.65	18000	21690.17	9509	21690.17	10512
Quito-318	21611.21	86400	21618.39	86400	21569.29	86400	21543.51	86400

The results with Istanbul instances do not help to distinguish between the alternative approaches since the corresponding problems are already small enough to be solved to optimality in less than 1 second while we verify that even the heuristic local branching may reach optimality. Looking at the results for Athens and Quito-122, we observe that both the LbDwCP algorithm and using the lazy constraints with Gurobi improve the performance of the commercial solver significantly. The local branching heuristic also provides quite satisfactory performance as it finds the optimal solution for Athens and almost optimal solutions for Quito-122 within the CPU time limit of 18000 seconds, one third of the CPU time required to find the optimal solution. With the largest instance, Quito-318, the solver terminates with an optimality gap of 4.39%. Given that the local branching finds solutions almost as good as the solver, and both LbDwCP algorithm and the lazy constraints implementation find even better solutions all within the same CPU time limit, it seems plausible to employ either the LbDwCP or the lazy constraints implementation both of which use constraint propagation also for large-scale instances. We conduct further experiments on instances of the same problem set with different demand patterns and varying the problem parameters such as the capacity of the vehicles, the size of the fleet and line service capacities.

5 Conclusion and Outlook

The multi-period version of the line planning problem in public transportation targets a more demand-responsive underlying line plan considering the sufficiency and timeliness of services on the PTN. We follow the footsteps of the developments in [6]; we present and discuss computational results for solution approaches that can be considered as alternatives to solving the problem directly with commercial solvers. Results show that it is still challenging to obtain optimal solutions for very-large instances but good-quality solutions can be obtained within reasonable time.

Further and ongoing research focuses on two challenges. First, solving LPsP to optimality requires more effort . Secondly, the accuracy of the multi-period approach can be further improved by avoiding approximations due to time-discretization.

References

1 Stefan Bunte and Natalia Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1(4):299–317, 2009.

2 George B. Dantzig, Delbert R. Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

3 Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98(1):23–47, 2003.

4 Johann Hartleb and Marie Schmidt. A rolling horizon heuristic with optimality guarantee for an on-demand vehicle scheduling problem. In Dennis Huisman and Christos D. Zaroliagis, editors, *20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020)*, volume 85 of *OpenAccess Series in Informatics (OASIcs)*, pages 15:1–15:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

5 Ulrich Pferschy and Rostislav Staněk. Generating subtour elimination constraints for the TSP from pure integer solutions. *Central European Journal of Operations Research*, 25:231–260, 2017.

6 Güvenç Şahin, Amin Ahmadi Digehsara, Ralf Borndörfer, and Thomas Schlechte. Multi-period line planning with resource transfers. *Transportation Research Part C: Emerging Technologies*, 119:102726, 2020.

7 Anita Schöbel. An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, 74:348–365, 2017.

8 Linus Schrage. *A Guide to Optimization-Based Multiperiod Planning*, chapter 3, pages 50–63. INFORMS, 2018.

9 Luis M. Torres, Ramiro Torres, Ralf Borndörfer, and Marc E. Pfetsch. Line planning on tree networks with applications to the quito trolebús system. *International Transactions in Operational Research*, 18(4):455–472, 2011.