MULTI-PERIOD LINE PLANNING PROBLEM IN PUBLIC TRANSPORTATION

by
AMIN AHMADI DIGEHSARA

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Doctor of Philosophy

Sabancı University
January 2022

MULTI-PERIOD LINE PLANNING PROBLEM IN PUBLIC TRANSPORTATION

Approved by:

Prof. Dr. Güvenç Şahin ....
(Thesis Supervisor)

Prof. Dr. Temel Öncan ....

Asst. Prof. Dr. Amine Gizem Tiniç ....

Prof. Dr. Dilek Tüzün Aksu ....

Prof. Dr. Tonguç Ünlüyurt ....

Date of Approval: January 27, 2022

ABSTRACT

MULTI-PERIOD LINE PLANNING PROBLEM IN PUBLIC TRANSPORTATION

AMIN AHMADI DIGEHSARA

Industrial Engineering PH.D. Dissertation, January 2022
Thesis Supervisor: Prof. Dr. Güvenç Şahin

Keywords: Public transportation planning, Line planning problem, Multi-period planning, Mixed-integer linear programming, Local branching, Benders decomposition, Logic-based Benders

decomposition.

Urban transportation systems deal with high fluctuations in demand over the day. To capture both temporal and spatial changes in transit demand, we propose a multi-period line planning approach. If such systems are also subject to limitations of resources, a dynamic transfer of resources from one line to another throughout the planning horizon should also be considered. A mathematical modeling framework is developed to solve the line planning problem with a cost-oriented approach considering transfer of resources during a finite length planning horizon of multiple periods. Given the NP-hard nature of the line planning problem, we first present a heuristic approach based on the generic local branching algorithm. We use real-life public transportation network data for our computational results. We conduct extensive computational experiments to demonstrate the efficiency of the algorithms. We show that the local branching algorithm significantly improves solution quality and computing time in comparison to the commercial solver. We also develop various Benders decomposition schemes to solve our multi-period line planning problem. As the traditional Benders decomposition does not show a promising performance, we resort to logic-based Benders decomposition which uses constraint propagation.

We demonstrate that the proposed logic-based decomposition outperforms the local branching algorithm; it is able to find high-quality solutions or medium and large instances. Finally, we present a second logic-based Benders decomposition with a smaller master problem while the subproblem is larger and more difficult to solve. We solve this challenging subproblem by reformulating it as a maximum flow problem; this decomposition produces a very effective solution method. Through computational experiments, we show that this algorithm performs better than all other approaches.

ÖZET

TOPLU TAŞIMA SISTEMLERINDE ÇOK DÖNEMLI HAT PLANLAMA PROBLEMI

AMIN AHMADI DIGEHSARA

Endüstri Mühendisliği, Doktora tezi, Ocak 2022
Tez Danışmanı: Prof. Dr. Güvenç Şahin

Anahtar Kelimeler: Kentsel Ulaşım Planlama, Hat planlama problemi, Çok dönemli Planlama., Karışık-tamsayılı doğrusal programlama, Yerel dallanma, Benders ayrıştırma, Mantık tabanlı Benders

Kentsel ulaşım sistemleri, gün içinde yüksek talep dalgalanmaları ile karşılaşmaktadır. talepteki hem zamansal hem de mekânsal değişikliklerle baş etmek için, hat planlama problemine çok dönemli yaklaşımı önerilmektedir. Eğer ilgili sistemin kaynakları da sınırlı ise, planlama ufku boyunca bir hattan diğerine dinamik bir kaynak aktarımının da göz önünde bulundurulmalıdır. Bu bağlamda, hat planlama problemini çok dönemli bir planlama ufku için kaynakların transferini de göz önünde bulundurarak maliyet odaklı bir yaklaşımla çözmek üzere bir matematiksel modelleme çerçevesi geliştirilmektedir. Hat planlama probleminin NP-zor doğası göz önüne alındığında, ilk olarak iyi bilinen yerel dallanma algoritmasına dayalı bir sezgisel yaklaşımı sunulmaktadır. Bilgisayısal sonuçlar için gerçek hayattan alınan toplu taşıma ağı verileri kullanıyoruz. Algoritmalarnın verimliliğini göstermek için kapsamlı bilgisayısal deneyler yapıyoruz. Yerel dallanma algoritmasının, ticari bir çözücüye kıyasla çözüm kalitesini ve hesaplama süresini önemli ölçüde iyileştirdiğini gösteriyoruz. Çok dönemli hat planlama problemimizin çözümü için çeşitli Benders ayrıştırma yaklaşımları da geliştiriyoruz. Geleneksel Benders ayrıştırması umut verici bir performans göstermediğinden, kısıt türetme yaklaşımı da kullanan mantık tabanlı Benders ayrıştırmasına başvuruyoruz. Önerilen mantık tabanlı Benders ayrıştırmanın yerel dallanma algoritmasından daha iyi bir performansa sahip olduğunu gösteriyoruz; mantık tabanlı Benders ayrıştırma, orta ölçekli ve büyük ölçekli problem örneklerinde iyi kalitede çözümler bulabiliyor. Son olarak, ana problem daha küçükken alt problemin daha büyük ve dolayısıyla çözülmesinin daha

zor olduğu ikinci bir mantık tabanlı Benders ayrıştırmasını sunuyoruz. Çözülmesi daha zor olan alt problemi, maksimum akış problemi olarak yeniden formüle ederek çözüyoruz; bu ayrıştırma çok etkin bir çözüm yönteminin ortaya çıkmasını sağlar. Bu algoritmanın diğer tüm yaklaşımlardan daha iyi performansa sahip olduğunu yaptığımız bilgisayısal deneylerle gösterebiliyoruz.

ACKNOWLEDGEMENTS

Firstly, I wish to express my deep gratitude to my distinguished supervisor, Prof.Dr. Güvenç Şahin, for his invaluable guidance, and for his friendly support and patience during my doctoral studies. It has been an honor and a privilege to have him as both my doctoral advisor and mentor. As well, I would like to thank Prof.Dr. Ralf Borndörfer and Dr. Thomas Schlechte for their technical assistance with this study. I also would like to express my gratitude to the Faculty of Industrial Engineering and to Professors Bülent Çatay, Barış Balcıoğlu, Ilker Birbil, Esra Koca, and Burak Kocuk. It is truly commendable the compassion, professionalism and friendly relationships that they establish with students, as well as the appealing atmosphere they maintain that encourages students to acquire knowledge.

As a member of my thesis committee, I wish to express my sincere appreciation to Prof.Dr. Tonguç Ünlüyurt, Prof.Dr. Dilek Tüzün Aksu, Prof.Dr. Temel Öncan, and Asst.Prof.Dr. Gizem Özbaygın Tiniç for their valuable suggestions and insightful comments.

It is my pleasure to express my gratitude to Sina Rastani for his unwavering support over the past five years. I appreciate your support, Sina. In addition, I would like to thank the members of the smart mobility lab teams, Simak Naderi Varandi, Merve Keskin, Ece Naz Duman, Nozir Shokirov, Saeedeh Ahmadi Basir, and Yasaman Karimian Hadi Ardebili, for their support as well as the enjoyable time we spent together in the lab.

The progress of my dissertation required more than academic support, and I am indebted to those who have listened to and, sometimes, tolerated my ramblings during the past five years. It is impossible for me to adequately express my gratitude and appreciation for their friendship. I wish to express my sincerest appreciation to Faraz, Nasim, Arash, Naeime, Ali, Mohammedreza, Sahand, Ehsan, Meisam, Mahsa, Vahid, Navid, Pegah, Farzad, Hamid, Sevda and Deniz for their unfailing support during my studies at the university. I would like to thank all of those who contributed to such memorable evenings out and about.

Furthermore, none of these accomplishments would have been possible without the support of my family. They gave me enough moral support, encouragement, and motivation to achieve my personal goals. As a testament to your support and encouragement, I am pleased to present my dissertation.

To my dearest parents,
for always loving and supporting me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The development of cities and, consequently, the rapid growth of the urban inhabitants have resulted in an ever-increasing demand for public transportation. In today's world planning and operating public transportation systems is particularly challenging for developing regions and metropolitan regions. Due to resource limitations, it is naturally necessary to maximize the efficiency of urban transportation by developing and maintaining demand-responsive and relatively sustainable systems. To address the relevant issues, researchers have been doing more research on urban planning problems in recent years. This thesis studies a new planning approach, new mathematical models and corresponding solution methods for the line planning problem (LPP), traditionally known as a strategic level decision-making issue in the context of public transportation planning.

We shall first provide some background on public transportation planning problems, particularly LPPs. We mainly discuss current mathematical models and respective solution approaches from the literature in order to pave the way for our primary motivation.

1.1 Public Transportation Planning

Public transportation systems consist of various transit alternatives such as buses, subways, and rapid transit railways developed at the local or regional level. These systems are available to the public and run at scheduled times. A public transportation network contains stations where passengers get on/off the vehicles and predefined paths which connect the stations. Public transportation agencies need to properly plan public transportation to ensure a sustainable system and mitigate inconvenience for passengers. In this respect, a vast range of planning problems needs to be unraveled beforehand. As shown in Figure 1.1, public transportation system planning has five distinct stages: network design (for infrastructure), line planning, timetabling, vehicle scheduling (rolling stock planning), and crew scheduling.

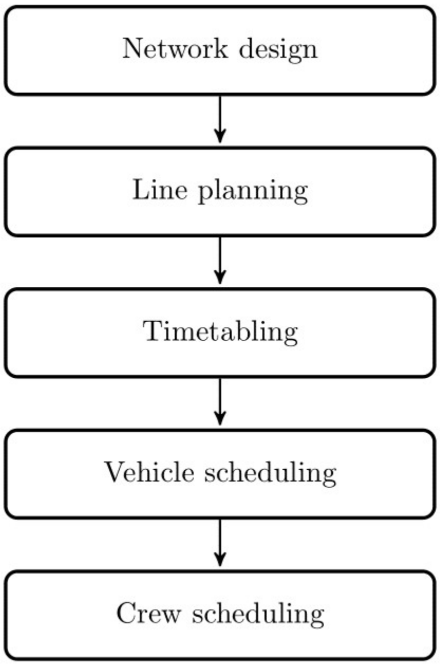


Figure 1.1 The planning steps in public transportation.

The planning process begins with designing a transportation network and sequentially proceeds by line planning at the strategic level; these two higher-level plans are followed by timetabling, and vehicle and crew scheduling which can be considered as tactical or even operational plans. However, with the sequential planning approach, the optimal solution of the prior stage may not even be feasible for the subsequent ones. On account of the drawbacks of the sequential planning approach, integrated planning approaches have become more popular and more viable in recent years. Different advanced techniques have been developed in order to find optimal or near-optimal solutions for various integrated planning strategies; they still are challenging and need further study. Despite these attempts, solving integrated planning models, particularly in real-life transportation systems, is an NP-hard problem; therefore, planning in public transportation mostly focuses on the sequential planning approach.

1.1.1 Network design for infrastructure

Transportation network design problem (TNDP) is the first stage of the five-staged sequential planning approach. TNDP is important to consider, due to the local population growth and development of new residential areas, and because of the high impacts on the subsequent planning stages caused. Studies for the TNDP are mainly classified into two categories: long-term strategic decision-making and predicting the passenger behaviors for the various strategies (Farahani, Miandoabchi, Szeto & Rashidi, 2013). Long-term strategic decisions are pertinent to the infrastructures of transportation networks, including stations, roads/rails (called edges) between stations, and lines. A line is a designated service by a vehicle on a path on the transportation network. In the context of the transportation network, stations with particular infrastructures are called terminals. In addition, to predict passenger behavior, an estimation model is considered in which a given demand matrix is assigned to a network, and predefined assessment factors are measured (van Nes, Hamerslag & Immers, 1988).

However, the state-of-the-art approaches formulate the TNDP as a bi-level problem of which the first level is concerned with the managerial decisions, and the second level mainly focuses on the passenger behavior (Farahani et al., 2013). The TNDP is an NP-hard problem (Ben-Ayed, Boyce & Blair III, 1988). Therefore, efficient solution approaches are required to obtain optimal or near-optimal solutions for real-life large-scale transportation systems. For a more detailed overview of the TNDP, see (Farahani et al., 2013).

1.1.2 Line planning

Given a public transportation network (PTN) and a line pool including all possible lines, the LPP finds the a subset of possible lines and determines how often service is provided along each selected line. In this respect, the developed line plan needs to meet the passenger demand and fulfill the operating limitations. It is important to create a line plan that maximizes passenger convenience, i.e., service quality, while the operational costs are small enough. Service quality copes with both reasonable trip time and fewer transfers for passengers, and shorter delays and headway time for vehicles.

There exist different alternatives for modeling the LPP. Since the LPP is the main topic of this thesis, we will introduce the pertinent mathematical models and the solution techniques which have already been employed successfully in Section 1.3.

1.1.3 Timetabling

Timetabling is at the tactical level of public transportation planning; the goal is to derive an efficient (periodic or aperiodic) timetable for transit services. A timetable specifies every arrival and departure of a vehicle at a particular station. If the departure and arrival events repeatedly run every given time units, the obtained timetable is called periodic; otherwise, it is an aperiodic timetable. One of the main reasons for periodic timetables is passengers' convenience, where it is easy to remember the departure/arrival time of vehicles. In the timetabling process, the events are coupled by activities through a so-called event-activity network. The event-activity network is a large graph with nodes representing an event corresponding to arrival and departure events, vehicle, and the station and directed edges representing activities that connect events. In the event-activity network, activities are categorized into three classes(Serafini & Ukovich, 1989):

- driving arc demonstrates the driving of a vehicle from one station to its next station,
- dwelling arc demonstrates a time duration when a vehicle stops at a station and passengers board/deboard, and
- transfer arc demonstrates a time duration when passengers/vehicles transfer between two lines.

There are different models and solution techniques for developing periodic and aperiodic timetables in the literature. Serafini and Ukovich Serafini & Ukovich (1989) present the first model for developing a periodic timetable which is the so-called periodic event scheduling problem (PESP). Kroon, Peeters, Wagenaar & Zuidwijk (2014); Liebchen & Möhring (2007) show that a timetable is feasible if the PESP solution is feasible. Although numerous scholars show that PESP may result in an efficient timetable, Odijk (1994) proves that the PESP is NP-complete. Therefore, they attempt to develop a mixed-integer problem formulation using optimization techniques including branch and bound (D'ariano, Pacciarelli & Pranzo, 2007), heuristic approaches (Goerigk & Schöbel, 2013), and stochastic optimization (Cacchiani, Caprara & Fischetti, 2012; Kroon, Maróti, Helmrich, Vromans & Dekker, 2008; Liebchen, Schachtebeck, Schöbel, Stiller & Prigge, 2010). On the other hand, the aperiodic timetable models are larger with respect to the number of variables; they may require a significant amount of computational effort to find optimal solutions. Branch and bound, Lagrangian relaxation, column generation, and dynamic programming are commonly used approaches. (Wang, Zhou & Yue, 2019; Yue, Wang, Zhou, Tong & Saat, 2016; Zhou & Zhong, 2007)

1.1.4 Vehicle scheduling

Following the timetabling stage, the vehicle scheduling stage determines the assignment of the vehicles to fulfill all the trips scheduled in the timetable(Ceder, 2011). Generally speaking, the vehicle scheduling problem finds the optimal schedule with a minimum total cost provided that

- each scheduled trip is assigned to a particular vehicle, and
- each vehicle carry out a feasible sequence of trips.

Vehicles are mostly considered as homogeneous (Freling, Wagelmans & Paixão, 1999). In general, a vehicle schedule contains vehicle blocks, and each block is given a vehicle. A vehicle block represents an operation of a vehicle that includes a continuous chain of trips that starts and ends in the depot. The main objective of the vehicle scheduling problem is to minimize the number of blocks that result in the minimum number of vehicles needed to cover all scheduled trips during the planning horizon (Borndörfer, Reuther, Schlechte & Weider, 2011; Maróti, 2006). Bunte & Kliewer (2009) provides a comprehensive review of models and solution approaches for the vehicle scheduling problem.

1.1.5 Crew scheduling

Crew scheduling is the final stage of the sequential public transportation planning approach. While chain of trips are assigned to the vehicles at the vehicle scheduling stage, crew scheduling involves assigning crews to trips, provided that each vehicle has a unique crew. There are two main planning tasks involving crew plan:

- tactical level decisions with a planning horizon of one year or longer, and
- operational planning with a planning horizon of a single day or a week.

In both levels, the crew scheduling problem deals with designing the efficient assignment for crews to meet all vehicle scheduling requirements for a given planning horizon (Caprara, Monaci & Toth, 2001; Şahin & Yüceoğlu, 2011; Vaidyanathan, Jha & Ahuja, 2007). Furthermore, crew scheduling problem needs to consider working time constraints or maximum working hours per day limitations. We refer the interested reader Heil, Hoffmann & Buscher (2020) for a detailed review of the framework of the crew scheduling problem.

1.1.6 Integrated planning approach

The integrated planning approach, which combines at least two of the planning stages, always outperforms the sequential approach (Schöbel, 2017). The integrated planning approaches aim to maximize passenger convenience and minimize the total operating cost associated with different planning stages, mostly ignored in the sequential planning approach. In this respect, Schöbel (2017) shows several integrated schemes and then formulates the problem as mixed-integer programming.

Kaspi & Raviv (2013) integrate line planning and timetabling to minimize the operational costs and user inconvenience. To decrease the user discomfort Karbstein (2016) discusses a novel integrated line planning and passenger routing problem which controls the unavoidable transfers in the line planning model. An unavoidable transfer happens when a passenger has to transfer among lines at least once. Liebchen & Möhring (2007) present an integrated line planning and timetabling problem to improve the line planning solution with respect to the number of direct passengers while finding the solution of timetabling. Goerigk, Schachtebeck & Schöbel (2013) provide an approach to assess the different line planning solutions and their effect on the timetabling. Zhang, Qi, Gao, Yang, Gao & Meng (2021) present an efficient mixed-integer linear programming problem formulation, in which the line planning and timetabling problems are optimized. They propose a bi-objective problem to minimize the total cost and travel time and convert it into a single-objective problem.

Integrated timetabling and vehicle scheduling have been addressed in several studies (Cadarso & Marín, 2012; Guihaire & Hao, 2008; Ibarra-Rojas & Rios-Solis, 2011; Petersen, Larsen, Madsen, Petersen & Ropke, 2013; Schmid & Ehmke, 2015). Haase, Desaulniers & Desrosiers (2001) develop an integrated model for solving the vehicle scheduling and crew planning simultaneously. Freling, Huisman & Wagelmans (2003) present an integrated optimization model for vehicle and crew scheduling. They propose a lagrangian relaxation where uses column generation to solve the problem.

Following the two-stage integrated models,Schöbel (2017) develops a bi-objective model for integrating the three planning phases: line planning, timetabling, and vehicle scheduling; she presents a mathematical formulation of an integrated model and discusses how iterative heuristic algorithms unfold the integrated problem. Alongside Schöbel (2017), Pätzold, Schiewe, Schiewe & Schöbel (2017) also study three consecutive planning stages in an integrated manner and solve the problem with different approaches. They also suppose vehicle scheduling in earlier planning stages.

1.2 Multi-period Planning Approach

Holt, Modigliani & Simon (1955) and Hanssmann & Hess (1960) are the earliest studies to report a multi-period planning approach; their motivation is the fluctuation in demand orders in a manufacturing environment. Holt et al. (1955) determine production quantities and also workforce levels for each month in a multi-year planning horizon. Hanssmann & Hess (1960) formulate the problem as a linear programming problem. Their formulations provide a foundation for many multi-period planning problems thus far. A multi-period approach has been discussed broadly in various problem domains including production planning and supply chains, facility location and resource allocation, and financial planning. In each problem domain, various uncertainties or variations over the planning horizon exist.

A number of multi-period models are developed in the field of public transportation, particularly in the area of timetabling and vehicle scheduling. Ibarra-Rojas, López-Irarragorri & Rios-Solis (2016) present a multi-period bus scheduling problem that determines the departure times for each bus line in order that each line has a particular planning period over the planning horizon. Along with the Ibarra-Rojas et al. (2016), the Guo, Sun, Wu, Jin, Zhou & Gao (2017) presents a multi-period timetable in metro transit to optimize transfer synchronization. Zäpfel & Bögl (2008) develop a multi-period vehicle scheduling model and crew scheduling model to combine tour and personnel planning. Kim & Kim (1999) a multi-period vehicle scheduling problem with homogeneous vehicles.

However, a considerable number of studies in the public transportation planning literature,in particular LPP, address the static single-period approach. Typically, the single-period strategy is a useful technique to consider general characteristics of public transportation systems in the absence of temporal and spatial variation in demand. However, demand in the real-world public transportation systems shows complex patterns in both time and space, which divides the planning horizon, e.g., a day, into periods. In this vein, a multi-period planning approach may be a good remedy for the non-steady aspects of the problem wherein the existing approaches are not responsive to challenging demand patterns.

1.3 Line Planning Problem

What widely began in the mid-1960s as an inquiry of public transportation planning has today come to incorporate the study of a wide range of problems in public transportation (W.Lampkin & P.D.Saalmans, 1967). LPP is a long-established problem in the context of public transportation systems. LPP is solved on a PTN which is composed of several stations and connections between stations. Passengers travel between pairs of stations. A line is a designated service by a vehicle on a path on the PTN. Based on a given PTN and travel demand for pairs of stations, the LPP seeks to find a set of lines together with their frequencies. The frequency of a line determines the number of times a service is repeated over the line within a period.

The introductory studies in the LPP date back to 1967 when W.Lampkin & P.D.Saalmans (1967) analyzed a real-world bus network. They propose a technique to operate the bus system with the minimum number of buses. They generate bus routes (lines) with an iterative heuristic approach, so-called the skeleton routes. A small set of routes is selected first, and sequentially new routes are added to cover uncovered stations. Frequencies are assigned to selected routes to maximize the service level. In addition, they propose a linear programming model to assign buses to trips. Silman, Barzily & Passy (1974) design a similar method in which selecting routes along with their frequencies is done in a two-phase process. In the first phase, they choose a route from a given candidate set by checking the operational cost and uncovered stations simultaneously. When the bus routes are defined, frequencies are assigned to routes such that the total travel time and discomfort are minimized. Rea (1971) presents a heuristic method that simultaneously establishes bus routes and frequencies. For the first time, Rea (1971) also considers the network design, called "template network", as one of the public transportation planning stages in the literature. Sharp, Jones & Bell (1974) discuss the planning process for bus routes and frequencies by formulating a capacitated fixed-cost multi-commodity transshipment problem. They develop a heuristic to allocate passengers to the routes. They find the most reasonable network by adding and deleting routes such that the operating costs and travel time are minimized. Hsu (1977), for the first time, presents a heuristic method that attempts to maximize the number of bus users.

In the 1980s, many planning problems were associated with bus transportation problems. However, due to governmental investments in intercity railway systems, there was a tendency toward using subways systems. A further increase in the passenger rate leads to having a non-efficient transportation railway system. In this respect, scholars tend to develop new formulations that may conceptually be similar to bus network problems but are still different in detail. Bussieck, Kreuzer & Zimmermann (1997) propose a mixed-integer linear programming formulation for the German railway network to find the optimal set of lines in which their primary objective is to maximize the number of direct travelers. Because of the size of the proposed network, they present a heuristic approach, define some valid inequalities, and use many network reduction techniques. They also find an upper and lower bounds to support the Branch and Bound to solve the relaxed version of the original problem in a reasonable time. The results demonstrate that their suggested heuristic approach works quite well in the presence of valid inequalities.

Claessens, van Dijk & Zwaneveld (1998) present a mathematical formulation for the railway line allocation. They minimize the operational cost of the Dutch railway system, contrary to the various studies, which maximizes the direct travelers. To solve the problem to optimality, they propose a branch-and-bound algorithm. Further, they define a set of binary decision variables to transfer the non-linear integer programming formulation to a linear. Finally, they compare the optimal solution against the solution found by the direct travelers strategy. Bussieck (1998) presents two linearization that have a smaller number of binary decision variables compared to Claessens et al. (1998). Many preprocessing techniques are utilized to develop a tight linear programming formulation which improve the lower bound provided in Bussieck et al. (1997). Bussieck (1998) consider both the German railway network and the Dutch railway system.

These two mathematical formulations and the proposed solution approaches create a general framework for future studies in the LPP, particularly for railway problems. Goossens, van Hoesel & Kroon (2004) present linear mixed-integer programming (MIP) to find the optimal line plan and assign frequencies to meet constraints. They expand the reduction techniques and the class of valid inequalities presented by Bussieck (1998) to solve large transportation networks, including the Dutch Railway Network. They report a significant improvement in the proposed branch-and-cut approach while applying their proposed extensions to the algorithm. Goossens (2004) extend the Claessens et al. (1998) and Bussieck (1998) formulations by considering different types of lines based on their halt (stop) pattern. Two alternative integer programming problem formulations based on the multi-commodity flow formulation are developed.The results show that the solution is better than the solution provided by single-type LPPs considered in the literature.

Despite all efforts to solve it in reasonable time, the LPP is still challenging even with all valid inequalities presented by Goossens (2004). In this respect, Torres, Ramiro, Borndörfer & Pfetsch (2011) consider a simple tree network to provide polynomial time algorithms. Surprisingly, they only find a few polynomial cases under some restricted assumptions. They show that the problem, even in a simple network, remains NP-hard. Borndörfer, Arslan, Elijazyfer, Güler, Renken, Şahin & Schlechte (2018) consider an integer mathematical model on Metrobüs system in Istanbul which has a line topology. Torres et al. (2011) consider only closed lines while Borndörfer et al. (2018) suppose that both closed lines and open lines.

In general, LPP models are categorized as passenger-oriented and cost-oriented according to their modeling approaches, while not all aspects of the LPP models are characterized by such classification Schöbel (2012). In the passenger-oriented approach, lines along with frequencies maximize the number of direct travelers or minimize the traveling time. In the cost-oriented approach, lines are selected to minimize the total cost, composed of operational and fixed costs. Most of the studies which have been discussed so far are cost-oriented models. However, in the passenger-oriented approach, Bussieck et al. (1997) maximize the number of direct travelers, while Puhl & stillerl (Puhl & stillerl), and Klier & Haase (2008) maximize the number of transported passengers. Rittner & Nachtigall (2009) and Schöbel & Scholl (2006) minimize the riding time, and Harbering (2013) minimizes the number of transfers. For a detailed review on line planning passenger-oriented mathematical models, see Schöbel (2012), Kepaptsoglou & Karlaftis (2009), and Farahani et al. (2013).

1.3.1 Passenger demand

Passenger demand plays an important role in public transportation planning. Demand is mostly demonstrated by a deterministic origin-destination matrix (OD matrix) considering a finite planning horizon or a fixed-length period. Each OD pair in the matrix shows the number of passengers who travel from an origin to a destination. There are various algorithms that calculate passenger demand (Borndörfer, Grötschel & Pfetsch, 2007; Kepaptsoglou & Karlaftis, 2009).

Demand often exhibits irregular fluctuations over a planning horizon because of holidays, work hours, etc. Some studies attempt to regulate high variations of demand in both time and space. In the context of optimal response to transport demand, two prominent strategies are introduced in the literature:

- forecasting the time-dependent travel demand,
- two-stage stochastic/robust optimization, and

Horváth (2012) proposes a model based on the passenger transfers to forecast the demand in time and space for the Hungarian railway network and generates a time-dependent OD matrix for public transport. Tsekeris & Tsekeris (2011) provides a comprehensive study on different approaches of demand forecasting in the transportation system. Jiang, Zhang & Chen (2014) propose a method for short-term prediction of demand under irregular fluctuations. However, it is expensive for transportation firms to develop a unique line plan for each period since it requires reviewing all subsequent stages (e.g., timetabling, vehicle scheduling, and crew scheduling) in public transportation planning.

Two-stage strategies typically deal with problems in the presence of data uncertainty. Stochastic optimization assumes that the exact probability distribution of uncertainty is realized beforehand, while robust optimization needs only an uncertainty set associated with data. Lusby, Larsen & Bull (2018) provide a comprehensive review on robustness in railway planning. They report that robust optimization is mostly used in the timetabling stage. (Pu & Zhan, 2021) develop a two-stage robust optimization approach under demand uncertainty for the Chinese high-speed railway system. Since their problem is hard to solve to optimality, a Lagrangian relaxation algorithm is introduced. The results show that their algorithm leads to a promising line plan. In addition, some studies present a two-stage stochastic programming model while the probability distribution of demand is known beforehand. More details on this topic can be found in An & Lo (2016) and Lo, An & Lin (2013).

Following to these strategies, Errico, Crainic, Malucelli & Nonato (2013) review a demand-responsive transportation system and propose a new framework under semi-flexible systems to classify various levels of hybridization of characteristics of both traditional public transportation systems and completely on-demand systems. It is clear that the highest level of demand responsiveness is attained at the one extreme where individual services are provided at the requested time and place. Both Malucelli, Nonato & Pallottino (1999) and Errico et al. (2013) claim that even the lowest level of responsiveness under such systems is reasonable only when transit demand is low and sparse. In the context of this thesis, we focus on traditional transportation systems and suppose that even the lowest demand level would benefit from a high degree of resource sharing both temporally and spatially.

1.3.2 Multi-period line planning problem

Demand for transportation, or transit demand, is the *sine qua non* of the LPP; simply put, there is no necessity for transport services when there is no demand. The traditional models for line planning consider a finite length planning horizon, e.g., a day, a certain part of the day, an hour. Accordingly, the demand during that planning horizon is considered irrespective of its timing; the problem is to find the line services and their frequencies to satisfy the demand in a steady state manner during the planning horizon. In other words, a static demand rate is assumed for the complete planning horizon. A comprehensive review on the LPP by Schöbel (2012) concludes by questioning the appropriateness of using the same line plan all over the day. One could easily unfold this question to assess the degree of fluctuation and variation in demand that requires the use of a non-steady handling. Borndörfer et al. (2018) note that the demand of the Istanbul Metrobüs system is extremely unsteady and asymmetric. They show that traditional line planning models are not convincing for these public transportation networks and discuss the lack of a modeling approach which adapts to the demand fluctuation during the planning horizon. When the problem environment or an aspect of the problem is not necessarily steady in time, multi-period planning and optimization arises as a remedy in the operations research literature (see Schrage (2018)).

Using the multi-period planning problem, it is possible to construct a more robust plan that identifies any potential infeasibilities that may arise in the subsequent phases of planning. Furthermore, we may also determine whether there is the possibility of assigning vehicles to operating lines at different times during the line planning phase, without having to construct a timetable or vehicle schedule. Developing a new formulation that integrates both of the above issues enables us to also address the two deficiencies in the literature, by integrating the fleet assignment problem with line planning and frequency planning. Several studies have been conducted on the optimization of timetabling and LPPs (Li, Xu & Han, 2019; Niu, Zhou & Gao, 2015; Yang, Han, Zhang, Han & Long, 2022). If the static LPP is replaced with the newly developed multi-period LPP (MPLPP), infeasible timetabling under dynamic demand may be eliminated with a pre-optimized line plan. Furthermore, the timetable that develops as a result of an MPLPP solution adapts more readily to fluctuating demand.

1.3.3 Vehicle Rotation

While a multi-period planning approach is a remedy for the non-steady aspects of the problem, the problem formulations become more complicated due to the constraints that are coupling the periods Schrage (2018). While constraints for a period allocate essential resources to activities, coupling constraints transfer the resources among activities from one period to the next. In this respect, a multi-period line planning problem comes inherently with resource constraints and their allocation throughout periods because a multi-period line plan needs to be feasible during each period with respect to associated resources as well. This necessity leads to integration of line planning decisions with decisions on allocation of resources. Integrations between different levels of decision making in transportation systems planning have been a more recent matter of interest in the literature. A detailed analysis on integration schemes of five planning stages in urban transportation systems is provided in Schöbel (2017). In the context of multiperiod line planning, we consider the decisions regarding with allocation of vehicles which can be considered part of the decisions made in vehicle scheduling of the five stages. In order to integrate the assignment and allocation of resources into the MPLPP model, we consider a single type of resource, namely vehicle rotation. Rotation of a vehicle refers to a process by which vehicles can transfer from one line to the other from one period to the next while minimizing the related total costs of the system throughout the planning horizon. The rotation of vehicles within the MPLPP over the planning horizons is one of the most important factors to consider since vehicles are one of the most limited resources in urban transportation systems. The vehicle rotations are discussed in detail in Section 2.3.

1.4 Thesis Organization

In chapter 2, we develop the first-of-its-kind multi-period model to solve the LPP with time-dependent demand. First, we demonstrate single-period LPPs (SPLPPs) as the foundation of the MPLPPs. Later, we consider resource allocation during the planning horizon, which enhances the usage of resources and, subsequently, the effectiveness of solutions. The computational result section introduces three real-life public transportation networks with different network characteristics. We observe that the multi-period planning approach outperforms the single-period planning approach. Further, we discuss the effect of the period length on the multi-period in line planning models with resource allocation. Finally, we show the computational difficulty in large network instances by presenting different parameter settings.

Chapter 3 discusses a local branching algorithm that can be scaled to solve MPLPPs with vehicle transfers, even for a very-large PTN. In Chapter 2, we show 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 in comparison to single-period static LPPs not only because of the convoluted structure of the MPLPP but also due to the integration of vehicle transfer constraints. Among the three real-world PTN examples, finding optimal solutions is not possible with a commercial solver, particularly for large networks. We use a local branching algorithm. Local branching is an iterative method which may provide a high-quality incumbent solution within an acceptable computational time. We discuss how local branching cuts divide the original problem into sufficiently smaller sub-problems. Then, we illustrate the implementation details and describe all cases arising by adding the local branching cuts to the problem. Finally, we present and discuss the computational results that verify the performance of the local branching algorithm as alternatives to solving the problem directly with commercial solvers. We further highlight the significant effect of choosing proper algorithm parameters in obtaining optimal or good-quality feasible solutions.

In Chapter 4, we first develop a classical implementation of Benders decomposition (BD). In this respect, we decompose our model into a master problem and a sub-problem, and we benefit from the totally Unimodular structure of the subproblem. To improve the convergence rate, we discuss many acceleration alternatives. Then, we provide exhaustive computation results and discuss the performance of the classical decomposition. Following the classical BD, two different logic-based BD (LBBD) approaches are employed. In the first LBBD, the original problem, by relaxing complicating constraints, is decomposed into a simple line planning master problem and a feasibility checking sub-problem. In the second LBBD, the original problem is divided into a generic multi-period LPP and a vehicle transfer feasibility problem by relaxing transfer decision variables and associated constraints. Since the latest decomposition approach has a considerable computational time, we consider the possible transfer through a bipartite graph and solve a maximum flow problem instead of solving the vehicle transfer feasibility problem. At each iteration, based on the information of the max flow, we generate a cut by helping the min-cut max-flow theorem. Finally, we present the complete computational results for proposed LBBD and analyze the results with the commercial solvers.

In the last part of this thesis, we conclude with remarks and present promising outlooks for further research.

2. PROBLEM DEFINITION AND MODEL

Urban transportation systems are subject to a high level of variation and fluctuation in demand over the day. When this variation and fluctuation are observed in both time and space, it is crucial to develop line plans that are responsive to demand. In order to plan and schedule a demand-responsive public transportation system, a multi-period line planning approach that considers a changing demand during the planning horizon is proposed. If such systems are also subject to limitations of resources, a dynamic transfer of resources from one line to another throughout the planning horizon should also be considered.

In this chapter, while our main goal is to discuss multi-period line planning, recognizing the connection between period-coupling resource constraints in multi-period planning and integration of line planning with planning of resources associated with later stages of planning, we take on both challenges by

- proposing a multi-period planning approach in response to high levels of fluctuation and variation in demand, and
- integrating the multi-period strategic-level line planning decision with resource constraints that ensure the availability and allocation of operational resources throughout the periods.

In the case of LPPs, an explicit consideration of time-varying demand in a multi-period setting is attempted for the first time. In the sequential planning approach, resources are considered in the last two stages. The consideration of resource availabilities in multi-period version of the problems in the first three stages calls for an integrated solution of these problems, which is a major open challenge in this area of research.

We lay the foundations of a multi-period line planning approach by introducing a multi-period line planning model in the form of an integer linear programming problem formulation. We extend this formulation with a consideration of resource allocation and resource transfer constraints in a multi-period setting and exemplify it with rotations of vehicles among lines. We present computational results to exhibit the value of and understand conditions that call for a multi-period approach in line planning. We also investigate the sensitivity of system characteristics and pose the period length determination as an inherent optimization problem.

In Section 2.1, we introduce the SPLPP and the MPLPP. We present mathematical models for both problems. In Section 2.2, we provide a comprehensive computational study of a multi-period line planning model for the three real-life public transportation systems. Mathematical models and computational experiments for multi-period line planning with resource allocation and transfers are presented in Section 2.3. In Section 2.5 we discuss the choice of the period length.

2.1 Single-period vs Multi-period Planning Approach

The most apparent difference between static and dynamic planning lies in the treatment of planning horizon. The static approach conceives the planning horizon as a single time period where the problem environment is static while dynamic planning considers the planning horizon as composed of several periods. Thus, the static approach can be considered as analogous to single-period planning. In contrast, the dynamic approach is analogous to multi-period planning. Accordingly, the main drawback of static or single-period modeling approaches in planning is the lack of ability of the models to capture the dynamic nature of the problem environment.

In the context of LPPs, demand is the most dynamic component of the problem environment. Although demand information is usually considered to be complex, it usually is accompanied with temporal information. However, the temporal aspect may be overlooked in order not to complicate the mathematics of the problem or the mechanics of the solution method. Losing the temporal information associated with the demand may overlook the dynamics of the system. LPPs are NP-Hard in generalSchöbel (2012); therefore, the problem is already challenging from a computational point of view. Hence, the traditional approaches rely on using a static single-period approach.

Recognizing the challenges in using static single-period line planning solutions to develop multi-period solutions, we are driven to consider a multi-period planning approach for the LPP. A single-period line planning solution supposes a simplified

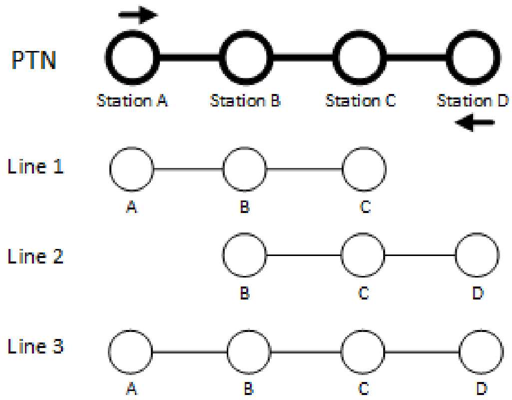


Figure 2.1 An instance with three lines and four stations.

base demand which may be the maximum or the average demand observed in an hour during a day. However, there are some shortcomings of these simplifying approaches:

- Considering the maximum demand for an O-D pair, undertaking a worst-case approach, may lead to unnecessary costs and low utilization of the system during the times of the day when demand is really low.

- With average demand, one is highly likely to find solutions with unnecessary high frequencies for periods with low demand. Moreover, it is possible that the demand of some O-D pairs during some periods may not be satisfied due to insufficient frequencies.

We first exemplify how the handling of time-dependent demand information may lead to undesirable solutions.

Example 1 *Figure 2.1 demonstrates a simple line topology network with four stations (A, B, C, and D) and three closed lines (i.e., in both forward and backward directions). An edge connects two adjacent stations. Without loss of generality, we assume travel times for all edges are equal. Line 1 covers stations A, B, and C; line 2 covers stations B, C, and D; line 3 covers all stations.*

The capacity of a vehicle is 50 passengers with 6 vehicles in the fleet. Vehicles are homogeneous. We consider a unit cost of 2 for lines 1 and 2, and a unit cost of 3 for line 3 as the rate per edge while we consider a cost of 10 as the fixed cost per line.

Table 2.1a and Table 2.1b summarize the passenger demand for OD pairs for two consecutive periods, where $t \in \{1,2\}$, respectively. There is at least one path on PTN for each OD pair. An OD pair is satisfied if all passengers are able to travel from its origin to destination.

Table 2.1 OD demand matrix for two consecutive periods

	Stations			
	A	B	C	d
A	0	50	100	125
B	25	0	175	150
C	50	75	0	150
D	25	200	50	0

(a) OD demand matrix for period t=1

	Stations			
	A	B	C	D
A	0	100	150	150
B	125	0	0	25
C	75	50	0	50
D	50	25	75	0

(b) OD demand matrix for period t=2

OD demand converts into edge demand by routing passengers through the shortest path beforehand. An edge demand indicates the traffic (travel) load on the edge. Since for each OD pair, only one path exists, we calculate an edge demand by summing up the number of passengers passes the edge within a period. Table 2.2a and Table 2.2b report the edges demand for two periods. A forward edge demand shows the travel demand in direction A to D, and a backward edge demand shows the travel demand in direction D to A.

Figure 2.2a and Figure. 2.2b show the optimal solution with the maximum demand approach and average demand approaches, respectively. The red color indicates the selected line. The bolder line shows the higher frequency. In the maximum demand approach, the frequency of line 1 is two, line 2 is three, line 3 is six, and the total variable cost is 28. On the other hand, for the average demand approach, line 1 with two frequencies, line 2 with one frequency, and line 3 with six frequencies are chosen while the variable cost is 24. Since all lines are selected in both approaches, the total fixed cost is 60. In the maximum demand approach, lines may use additional frequencies in some periods. Contrarily, with the average demand approach, demand may not be satisfied because of a service shortage. Obviously, in both cases, the algorithms do not provide a promising line plan.

Further, we solve the problem for each period separately. In the solution with the period-by-period approach, in the first period, the frequency of line 2 is five, and line 3 is six, while the variable cost is 22. In the second period, the frequency of line

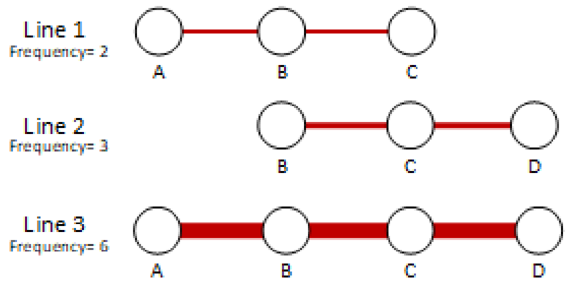
Table 2.2 Edges demand for two consecutive periods.

	Line direction	
	Forward	Backward
(A, B)	400	250
(B, C)	550	350
(C, D)	375	275

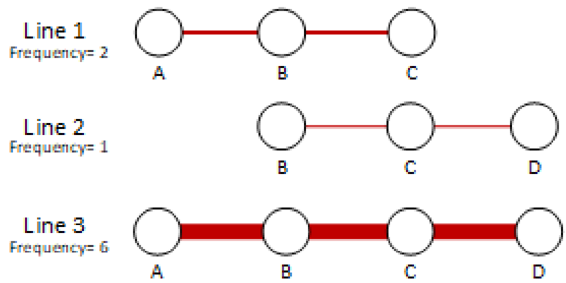
(a) Edges demand for period t=1.

	Line direction	
	Forward	Backward
(A, B)	337.5	175
(B, C)	437.5	275
(C, D)	325	212.5

(b) Edges demand for period t=2.



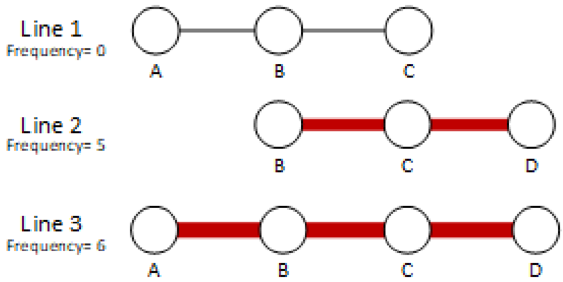
(a) The solution with the maximum demand approach with a total cost of 116.



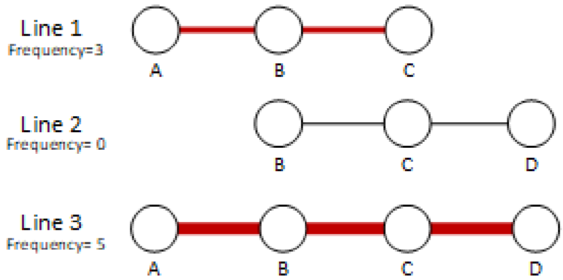
(b) The solution with the average demand approach with a total cost of 108.

Figure 2.2 The solution for the maximum demand and average demand approach.

1 is three, line 3 is five, and the variable cost is 21. Since two lines are selected, the fixed cost is 20. Figure 2.3 demonstrates the optimal solution for both periods with the period-by-period approach.



(a) The solution for t=1 with a total cost of 48.



(b) The solution for t=2 with a total cost of 41.

Figure 2.3 The solution for the period-by-period approach with a total cost of 83.

The performance of different approaches is summarized in Table 2.3. The first column shows the name of the approach. The second column indicates the number of periods, consisting of two periods. Column s 3-5 show the fixed, variable, and total costs. The next column shows the average utilization of vehicles, and the last column indicates the number of possible unsatisfied demands.

Results show that the total cost of the period-by-period approach is lower than the other two approaches. The average utilization of two periods in the period-by-period approach is better than the others. Besides, in the solution with the average demand approach, the demand on edges (B, C) and (C, D) are not completely satisfied in the first period. Generally, lower utilization of vehicles and unsatisfied demand are two significant drawbacks of maximum and average demand approaches. As a result, solving an LPP period-by-period may lead to a better result. However, in the period-by-period approach, we have to solve the problem once for each period which may be expensive regarding the computation time.

2.1.1 Single-period line planning problem

Table 2.3 The results of the three approaches in example 2.1.

Planning approach	Period	Fixed cost	Variable cost	Total cost	Average utilization of vehicles	Number of edges with unsatisfied demand
Period by period	1	20	28	48	0.75	0
	2	20	21	41	0.79	0
Maximum demand	1	60	56	116	0.84	0
	2				0.69	0
Average demand	1	60	48	108	100	2
	2				0.78	0

Despite all the known shortcomings we mentioned earlier, traditional approaches rely on using static single-period approaches because the problem is already challenging from a computational point of view. In this section, we introduce the SPLPP.

An LPP includes a directed network $PTN = (V, E)$ where V is the set of stations and E is the set of directed links joining each pair of stations. d_e denotes the number of passengers to travel on edge $e \in E$. L is the set of predetermined lines that are simple paths on the PTN. For each $e \in E$, $L_e \subseteq L$ is the set of lines which includes edge e and cover its demand. A line is said to serve all the transport links (corresponding to edges in E) between consecutive stations on the path from its starting station to the terminal station and covers part of the demand over those edges. It is not always possible to use any station as starting or terminal for a line since certain infrastructural or technological features may be required for such stations. Therefore, the number of lines is almost always limited in practice. However, it is also possible to enumerate all possible pairwise combinations of such (starting and terminal) pairs to make up the predetermined set of lines. In the scope of a cost-oriented optimized line plan, two types of costs are considered: a fixed cost associated with the usage of a line, c_l^f, and rate of the variable cost depending on the size of the service capacity offered to the passengers, c_l^o, usually proportional to the length of a line $l \in L$. In this respect, an SPLPP is defined as follows.

Definition 1 *Given a predefined set of lines L, an SPLPP seeks a subset of lines $L^* \subset L$ along with their frequencies to minimize the total costs and serve the passengers demand.*

In an integer programming problem formulation for the SPLPP, an integer decision variable v_l shows the frequency of line l. The binary decision variable y_l is equal to 1 if line $l \in L$ is selected; it is equal to 0, otherwise. If the passenger capacity of a vehicle is $\mathcal{K}$, the maximum number of vehicles in the system is $\mathcal{U}$, and the number of vehicles limitation on a line in a period is $\mathcal{W}$, an integer programming problem formulation for the SPLPP becomes

$$\text{(2.1)} \quad \text{minimize} \quad \sum_{l\in L} c_l^f y_l + \sum_{l\in L} c_l^o v_l$$

$$\text{(2.2)} \quad \text{subject to} \quad \sum_{l\in L_e} \mathcal{K} v_l \geq d_e \qquad \forall e \in E,$$

$$\text{(2.3)} \quad \mathcal{W} y_l - v_l \geq 0 \qquad \forall l \in L,$$

$$\text{(2.4)} \quad \sum_{l\in L} v_l \leq \mathcal{U}$$

$$\text{(2.5)} \quad y_l \in \{0,1\} \qquad \forall l \in L,$$

$$\text{(2.6)} \quad v_l \in N \qquad \forall l \in L.$$

The objective function (2.1) minimizes the total cost which is composed of the variable cost and the fixed cost. Constraints (2.2) make sure that the demand d_e on each $e \in E$ is covered by the total capacity of the lines covering the edge while the set of such lines is denoted by L_e. Constraints (2.3) ensure that vehicles are assigned to a line only if the line is selected. Constraint (2.4) restricts the total number of available vehicles. Finally, (2.5) and (2.6) are domain constraints for decision variables.

2.1.2 Multi-period line planning problem

An SPLPP formulation provides a foundation to derive an MPLPP formulation through discretization of time. Our simple example in Section 2.1 demonstrates that simplifying the problem to use solutions of SPLPPs may not necessarily be optimal, if not feasible. Even with the period-by-period approach that requires solving a sequence of SPLPPs, the combined solution may be feasible only locally for each period as the usage of resources in the system, and the flow of passengers are not coordinated over the periods. Accordingly, we propose a multi-period planning approach to develop a line plan that does not only consider the change in demand in different periods but also provides a solution for the coordination of system-wide resources throughout the planning horizon.

Definition 2 *Given a predefined set of lines L, and a planning horizon, T, which is divided into discrete time period, $t \in T$, of a predetermined length, an MPLPP finds a subset of lines $L^* \subset L$, as well as their frequencies in each period $t \in T$ to* *minimize the cost such that the total service capacity of all lines serving an edge is sufficient to cover the edge's travel demand.*

In an integer programming problem formulation forthe MPLPP, d_e^t denotes the number of passengers to travel on edge $e \in E$ in period $t \in T$. v_l^t is a non-negative integer variable that denotes the frequency provided on line l in period $t \in T$. The integer programming problem formulation for the MPLPP is

$$\text{minimize} \quad \sum_{l \in L} c_l^f y_l + \sum_{l \in L} \sum_{t \in T} c_l^o v_l^t \tag{2.7}$$

$$\text{subject to} \quad \sum_{l \in L_e} \mathcal{K} v_l^t \geq d_e^t \qquad \forall e \in E, \forall t \in T, \tag{2.8}$$

$$\mathcal{W} y_l - v_l^t \geq 0 \qquad \forall l \in L, \forall t \in T, \tag{2.9}$$

$$\sum_{l \in L} v_l^t \leq \mathcal{U} \qquad \forall t \in T, \tag{2.10}$$

$$y_l \in \{0,1\} \qquad \forall l \in L, \tag{2.11}$$

$$v_l^t \in N \qquad \forall l \in L, \forall t \in T. \tag{2.12}$$

In the objective function (2.7), the fixed cost is the same as in $SPLPP$ while the variable cost is multiplied by the use of vehicles in each period. Constraints (2.8) make sure that the number of travelers transported on an edge during each period must not exceed the total capacity of the lines covering the edge. Constraints(2.9) and (2.10) are similar to constraints (2.3) and (2.4).

2.2 Computational Experiments for MPLPP

The aim of our computations is to demonstrate how the solutions change with the use of a multi-period planning approach and understand which parameters of the problem have a significant impact. For this purpose, we consider three cases

- The Istanbul Metrobüs system, as our motivating example, experiences a high level of fluctuation in demand both temporally and spatially. Since the network topology of the Metrobüs system corresponds to a simple path, the inherent computational difficulties are not experienced in practice. It allows us to focus on multi-periodicity aspects with a very detailed analysis.

- The topology of the Athens Metro infrastructure network is not a special case, yet a centralized and quite simplistic network. It is neither small nor large with respect to the size of the network and the number of lines.
- The infrastructure network of the Trolébus system in Quito is a tree. However, the network is quite large and the number of lines considered in the problem is high.

While choosing these instances, we first intend to exemplify our findings by using optimal solutions of the problem that can also be obtained in a reasonable time. As a result, all instances of the Istanbul Metrobüs system can be solved to optimality under all settings due to the size of the network and the size of the line set, and the analysis can be done by comparing optimal solutions rather than near-optimal solutions. It is partly achieved for the case of Athens while it exemplifies how the period length may be the source of a computational challenge. The Quito instance demonstrates that the MPLPP can be much more difficult in practice when compared to the SPLPP counterpart.

2.2.1 Results of MPLPP for the Istanbul metrobüs system

Istanbul Metrobüs system The Metrobüs is a bus rapid transit system that provides a backbone for the public transportation system of Istanbul with connections to underground rail, bus, and light rail. It has 44 stations from Beylikduzu on the far-west of the European land of Istanbul and Sogutlucesme on the Asian land; the map in Figure 2.4 shows the geographical positioning of the system.

Figure 2.4 Istanbul map showing the Metrobüs system

BRT systems are known and popular for providing fast service. According to Basso et al. Basso, Feres & Silva (2019), 170 cities around the world have BRT systems covering 376 corridors and a distance of 5,046 km. It is also well-known that such systems usually suffer from excess demand as in the Istanbul case. Currently, the Metrobüs system works with 9 closed lines (34, 34T, 34BZ, 34U, 34Z, 34C, 34A, 34AS, 34G) as shown in Figure 2.5. The shaded area between Zincirlikuyu and Bogazici Koprusu shows the inland water Bosphorus while the other one between Ayvansaray and Halicioglu is the Golden Horn, a primary inlet of the Bosphorus, which hosts the ancient harbor of Istanbul. Table 2.4 presents basic information on lines including starting and ending stations along with the length of each line.

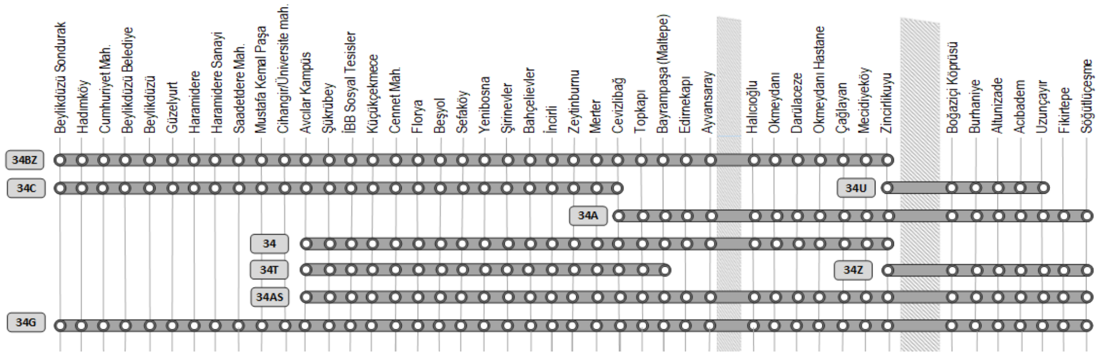
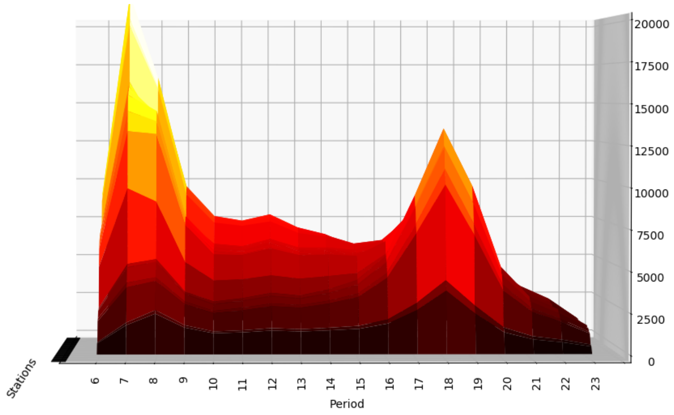


Figure 2.5 Network map of Metrobüs system with terminals and lines.

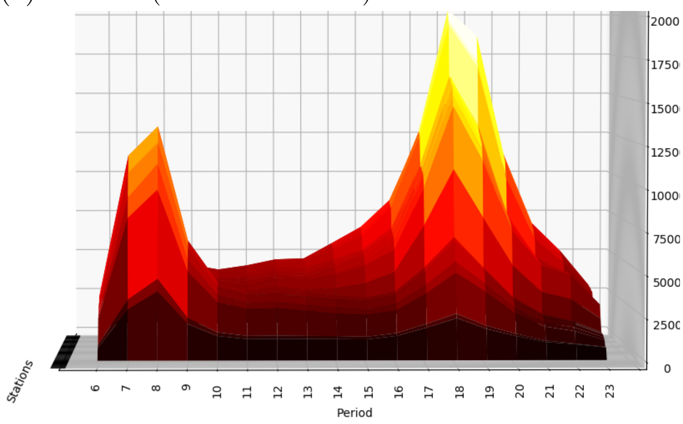
Table 2.4 Information on the lines used in Metrobüs system.

Line	**Starting station**	**Ending station**	**Length (in meters)**
34	12.Avcilar Kampus	37.Zincirlikuyu	29900
34A	26.Cevizlibag	44.Sogutlucesme	22600
34C	1.Beylikduzu Sondurak	26.Cevizlibag	28600
34G	1.Beylikduzu Sondurak	44.Sogutlucesme	51200
34U	37.Zincirlikuyu	42.Uzuncayir	9400
34T	12.Avcilar Kampus	28.Bayrampasa	19200
34Z	37.Zincirlikuyu	44.Sogutlucesme	11300
34AS	12.Avcilar Kampus	44.Sogutlucesme	41200
34BZ	1.Beylikduzu Sondurak	37.Zincirlikuyu	39900

Demand data for all O-D (station) pairs covering a planning horizon of one day is provided for periods of 1-hour length from 6 am to midnight (corresponding to 18 time periods). It is known that the number of passengers who travel on the network changes drastically depending on the time of the day and day of the week. We have three different daily demand data: an average weekday (denoted as Weekday hereafter), Saturday, and Sunday. Each daily data exhibits a high level of variation and asymmetry in time. In order to show the load on the network, we first convert O-D demand into edge demand, and show it both spatially and temporally. Figure 2.6 displays two charts showing the amount of edge demand in both forward (from west to east) and backward directions. The peak and the off-peak periods are easily observed: between 7 am to 9 am traffic demand is high in the direction from east to west and 5 pm to 8 pm from west to east.



(a) Forward (from west to east) direction



(b) Backward (from east to west) direction

Figure 2.6 Average weekday demand in Metrobüs system

In our models, the objective function includes a fixed cost for selecting a line and a variable cost for operating it. We suppose that fixed costs are mostly related to the cost of terminal stations. At a terminal station, additional space is needed for concourse or vehicle transfers, and terminals are usually facilitated with extra equipment. We assume that when a line is selected, the fixed cost should be charged once for the complete planning horizon. We also suppose that variable cost is proportional to the line length. In the computations, we consider a unit cost of 1 as the rate per 1 km of a line while we consider a cost of 180 as the fixed cost per line (by considering haphazardly 10 times of the unit operational cost per km of a vehicle trip and multiplying it with the number of periods (18) in the planning horizon).

We consider the Metrobüs system with a planning horizon of one day. Since the demand data is provided for 1-hour periods; the planning horizon is divided into 18 periods of 1-hour length since the planning horizon considers the day from 6 am to midnight.

In the demand data, the largest demand figure is 2158.8 in the Weekday data set while it is 688 and 390, respectively, for Saturday and Sunday. Besides, the average demands are 18.03, 15.14, and 11.05 respectively, for Weekday, Saturday, and Sunday. For the baseline computations, parameters of the systems are set as follows: the capacity of a vehicle is 250 passengers ($\mathcal{K} = 250$) with 200 vehicles in the fleet ($\mathcal{U} = 200$) and a maximum number of 36 vehicles to be assigned to a line ($\mathcal{W} = 36$). The set of candidate lines among which the lines to be operated are selected is limited to the existing 9 lines.

All computational experiments are carried out on a computer with Intel Core(TM)i5-6200 CPU v2 2.30 GHz CPU and 4 GB RAM, using Gurobi Optimizer 7.5.2 as the integer programming solver with Python 3.6.2. All reported solutions are optimal.

2.2.1.1 Single-period approach vs. multi-period approach

To begin with, we should test the value of a dynamic multi-period planning approach against a static single-period approach. For this comparison, we solve a 1-day problem with the MPLPP formulation for once. Alternatively, we solve the problem of each 1-hour period separately with the SPLPP formulation and combine the solutions of 18 periods to make up a 1-day solution. While combining, we recalculate the fixed cost component by charging the fixed cost for each line only once in case a line is selected in more than one period.

Table 2.5 shows the results for the three daily instances (Weekday, Saturday, and Sunday) in terms of five solution metrics:

- Total cost is the sum of the fixed costs and operating costs; it is directly the value of the optimal objective function of MPLPP while it is recalculated for SPLPP to avoid multiple charges of the fixed cost for the same line.
- Total frequency is the total number of services/trips to run during the planning horizon.
- Distinct lines correspond to the number of lines selected for the complete planning horizon.
- Line usage shows the sum of the number of times each line is used in 18 periods.
- Distance traveled shows the total distance traversed by all vehicles on all lines.

According to the results in Table 2.5, the total cost of the multi-period approach is significantly lower than that of the single-period approach in all three instances although the total frequency and line usage are always higher with MPLPP. Single-period solutions need to travel longer distances and operate more lines to satisfy the same demand. This is, indeed, the underlying reason for the sub-optimality of combined single-period solutions when compared to multi-period solutions. In addition, we also observe that most of the services are run on lines with shorter lengths in a multi-period approach.

Table 2.5 A comparison between SPLPP and MPLPP ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

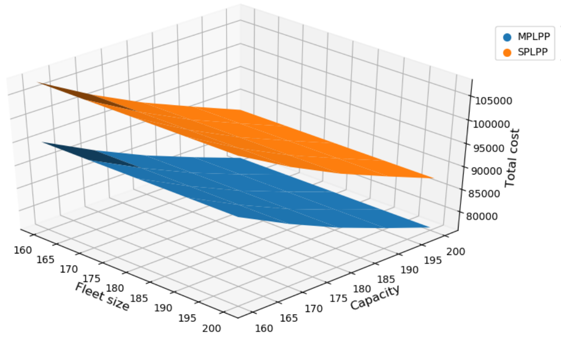
	Weekday		Saturday		Sunday	
	SPLPP	MPLPP	SPLPP	MPLPP	SPLPP	MPLPP
Total cost	64,058.80	61,334.60	51,731.40	48,807.00	37,636.80	35,377.80
Total frequency	875	1,070	635	822	457	569
Distinct lines	7	5	6	5	6	5
Line usage	49	90	43	90	38	90
Distance traveled	62798.80	60434.60	50651.40	47907.00	36556.80	34477.80

To complete our comparison, we expand this setting for various values of the system parameters. The baseline computations presented in Table 2.5 are repeated for $\mathcal{K} = 250$, $\mathcal{U} = 200$ and $\mathcal{W} = 36$. Figure 2.7 shows the total cost for settings where $\mathcal{K} \in [160, 200]$ and $\mathcal{U} \in [160, 200]$ for $\mathcal{W} = 30$ and $\mathcal{W} = 45$. Apparently, for every possible setting with a feasible solution, SPLLP solutions are costlier than MPLPP solutions. At the same time, we observe that the total cost is higher when resources are more limited, i.e., when the vehicle capacity is small and the number of available vehicles is low. We, therefore, postulate that MPLPP provides substantially better solutions in terms of cost when compared to combined solutions of SPLPP. In both charts with different $\mathcal{W}$ values, the results are the same.

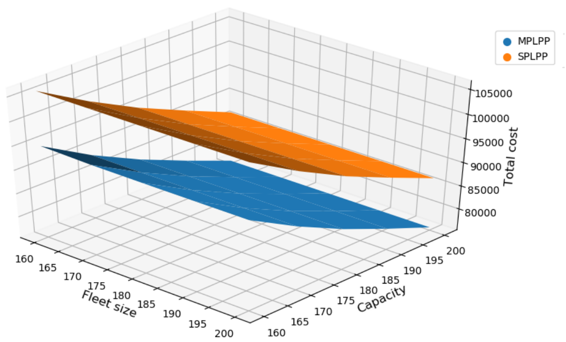
2.2.1.2 Level of variation in demand

In order to understand how sensitive the multi-period approach is to fluctuations in demand, we shall investigate the changes in optimal cost in response to changes in the level of the variation in demand. Based on reference O-D demand data, new demand data are generated by either increasing or decreasing the variation with respect to time.

For a given O-D pair, the average demand over all periods is calculated first. Then, for any period with a demand above the average, the demand is increased by a



(a) $\mathcal{W} = 30$



(b) $\mathcal{W} = 45$

Figure 2.7 Total cost with SPLPP and MPLPP for combinations of $\mathcal{K}$ and $\mathcal{U}$.

fraction of the difference from the average while it is decreased by the same fraction of the difference from the average for periods with a demand below the average. With this modification, a demand data with more fluctuation and variation in time is generated for a given modification fraction. When the opposite is done, i.e., an increase for a period below the average and a decrease for a period above the average, a demand data with less variation is obtained. We use 25% and 50% as modification fractions, and obtain four new data set by changing the variation in both directions.

Table 2.6 shows two variation-related statistics for all three daily instances. Range, denoted by δ, shows the difference between maximum demand and minimum demand among all demand figures (over all O-D pairs and periods). Both range and standard deviation, denoted by σ, decrease (increase) as the variation of demand decreases (increases). It should also be noted that with this modification, the total demand over all O-D pairs and the total demand for a particular O-D pair are approximately the same. The modification alters only the distribution of passengers in time for a given O-D pair.

For all five demand data sets of all three daily instances, the change in optimal total cost is demonstrated in Figure 2.8. The figures on the bars show the percentage difference with respect to reference (representing the original data). It is easily and

Table 2.6 Statistics of demand data for different levels of variation.

	Weekday		Saturday		Sunday	
Modification fraction	δ	σ	δ	σ	δ	σ
50% (decrease)	1079.40	23.27	344.00	14.94	195.00	10.77
25% (decrease)	1619.10	34.92	516.00	21.74	292.50	16.16
- (reference)	2158.80	46.56	688.00	28.99	390.00	21.55
25% (increase)	2698.50	58.19	860.00	36.23	487.50	26.94
50% (increase)	3238.20	69.83	1032.00	43.48	585.00	32.32

clearly observed that the optimal objective function value increases when the level of demand variation increases. It should again be noted that for a given daily instance, the total demand for each O-D pair is approximately the same for all five demand data sets. Therefore, the cost of an optimal line plan is clearly sensitive to the temporal variation and distribution of demand in time over a fixed length planning horizon (for a given total demand). These results manifest the need and significance of multi-period approaches in the scope of LPPs in response to high levels of demand variation and fluctuation. In addition, we also observe that the effect of variation is more evident when the total demand is higher as for the Weekday instance. This asserts that the effects of variation and fluctuation are even more significant for overly-crowded systems as exemplified by the Istanbul Metrobüs.

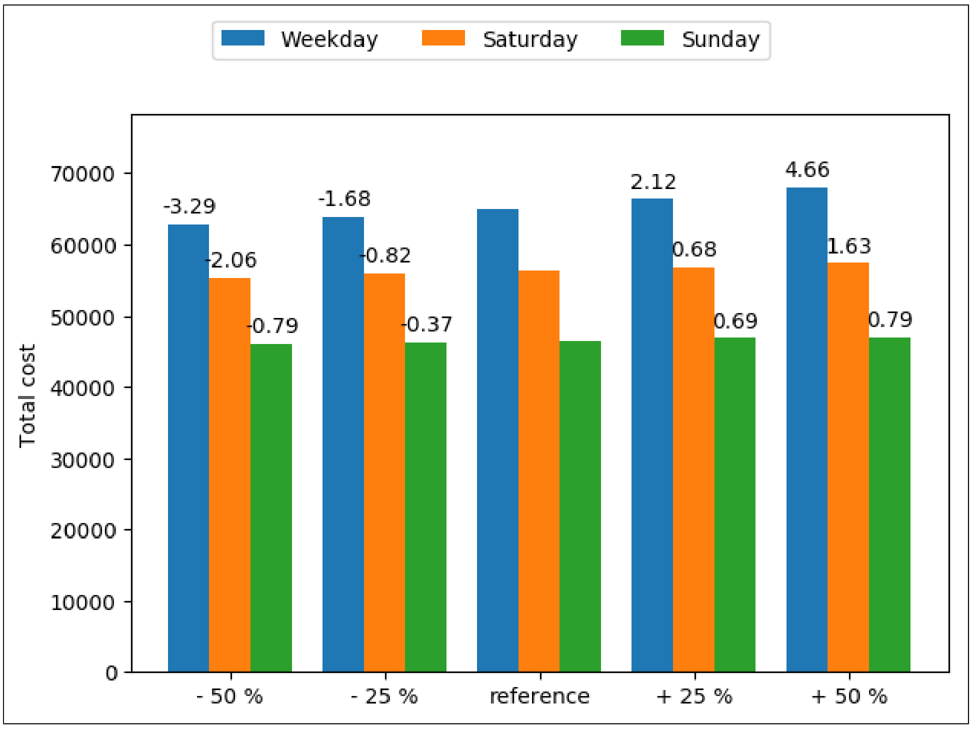


Figure 2.8 Optimal total cost values for demand data with different levels of variation for all daily instances.

While investigating the effect of demand variation, we shall also avoid the potential bias due to system parameters such as vehicle capacity, fleet size, and the limit on the number of assigned vehicles to a line.

2.2.1.3 Size of the line set

In our computational experiments, the set of candidate lines is limited to the existing lines. However, this set can be expanded by enumerating all possible lines based on the current terminal stations used by the existing lines. This would make 45 lines, adding 36 to the existing 9 lines. This will affect the size of the problem and the solution time of the solver. The results for MPLPP showing the solution time along with other metrics are in Table 2.7.

Table 2.7 Comparison with respect to size of the candidate line set ($\mathcal{K} = 250, \mathcal{U} = 200, \mathcal{W} = 36$)

	Weekday		Saturday		Sunday	
	9 lines	45 lines	9 lines	45 lines	9 lines	45 lines
Total cost	61,334.60	61004	48,807.00	48417.00	35,377.80	35124.60
Total frequency	1,070	1037	822	628	569	531
Distinct lines	5	6	5	5	5	5
Line usage	90	108	90	89	90	89
Distance traveled	60434.60	59924.0	47907.00	47517.6	34477.80	34224.6
Solution time	3.56	5753.48	7.40	5068.34	5.21	5060.20

We observe that an increase in the size of the set of candidate lines from 9 to 45 has a limited effect on the total cost (around 1% for Weekday data) while it yields a significant inflation in computational effort (around 1000 times of the original solution time). Accordingly, it is reasonable to conduct the analysis with the current set of lines for the sake of computational effort. On the other hand, even this limited experiment shows that the computational complexity is a potential issue for further research on multi-period planning approach.

2.2.2 Results of MPLPP for the Athens metro

We conduct additional computations on the Athens Metro data available in LinTim toolbox Harbering, Schiewe & Schöbel (1999). This system was studied earlier in Siebert & Goerigk (2013), Schmidt & Schöbel (2015), Pätzold & Schöbel (2016) and Manitz, Harbering, Schmidt, Kneib & Schöbel (2017) for various aspects of public transportation systems including line planning and timetabling. The network has 51 stations and 52 transportation links (almost-tree with 2 circuits at the center of the network); 59 lines carry the passengers for 2385 OD-pairs. The demand data in LinTim considers only one period as expected. In order to generate a multi-period demand; we use a truncated normal distribution for each O-D pair by assuming the given demand figure as the mean of the distribution and a variance calculated over all O-D pairs. Accordingly, based on this distribution, we generate 18 periods of demand data representing the data for a planning horizon of one day without any particular pattern. For the all computations in this section, we use the same data generation parameters as in the Istanbul Metrobüs system and set the systems parameters as $\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$.

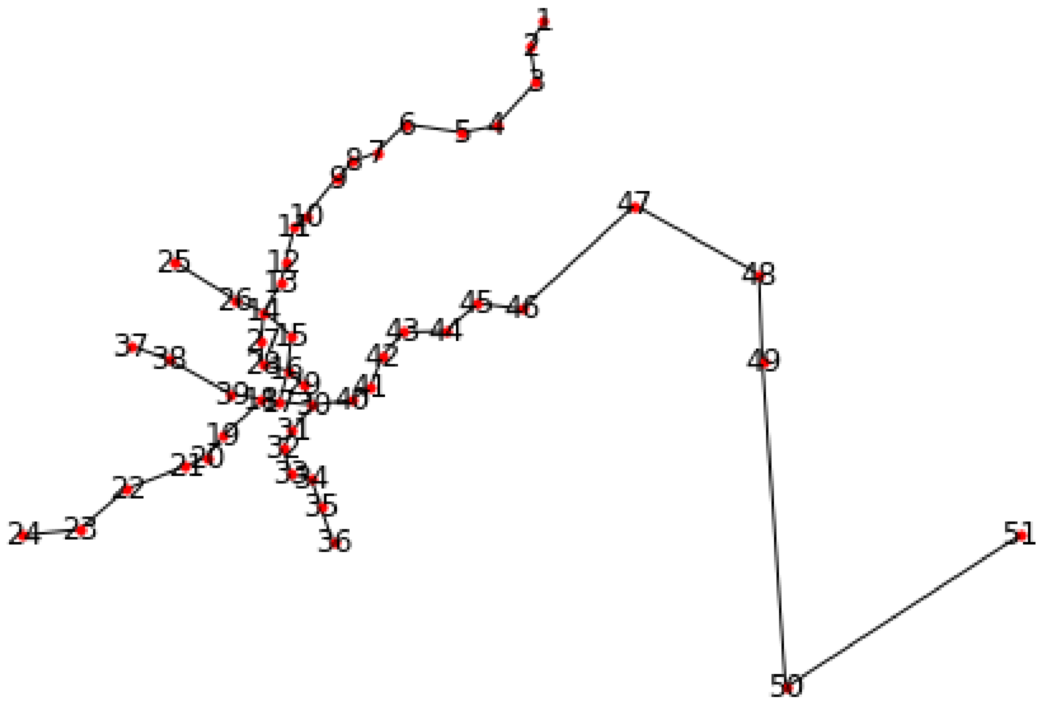


Figure 2.9 Network map of Athens metro.

As in Section 2.2.1.1 for the Istanbul Metrobüs system, we first compare results of SPLPP and MPLPP to see how a dynamic multi-period planning approach changes the solutions obtained from single-period approach. The results are summarized in Table 2.8. For all but the number of distinct lines, the comparisons are similar to those in Table 2.5. The total cost of the multi-period solution is lower than the single-period and oppositely, the total lines in use and frequencies are higher in multi-period in the Athens instance. One can quite easily observe that shorter lines are preferred in this optimal solution.

Table 2.8 A comparison between SPLPP and MPLPP ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

	SPLPP	MPLPP
Total cost	74536.31	68030.58
Total frequency	3208	3214
Distinct lines	7	16
Line usage	126	288
Distance traveled	659486783.97	65150583.52

The difference in terms of the total cost between multi-period solution and single-period solution is significant. Therefore, we have enough reason to believe that the level of variation in demand may also have an effect on the multi-period solutions of the problem. Table 2.9 shows the statistics for the newly regenerated demand data for the Athens instance in the same fashion as we present in Table 2.6 for the Istanbul Metrobüs instance. In Figure2.10, we show how the cost of the optimal solution changes with respect to level of variation while the figures on the bars show the percentage difference with respect to reference. While the cost is almost stable for lower levels of variation, it increases drastically when the demand variation increases similar to what we observe in the Metrobüs. The slight change with lower levels of variation is negligible since the total demand may increase or decrease slightly due to rounding while our method is arranged to keep the total demand more or less the same. Even though, these results confirm how the total cost increases when the variation in demand increases. The results in the Athens instance, as in Section 2.2.1.2 in the Metrobüs instance, verifies the significant effect of fluctuations of demand on multi-period solutions.

Table 2.9 Statistics of demand data for different levels of variation.

Modification fraction	δ	σ
50% (decrease)	351.78	22.41
25% (decrease)	527.00	33.61
- (reference)	703.56	44.51
25% (increase)	873.35	55.73
50% (increase)	1043.14	66.41

2.2.3 Results of MPLPP for the Quito trolébus

The Quito Trolébus system together with the feeder lines constituting a single path and 14 branches has a tree topology; we use it as a large-scale instance to demonstrate how MPLPP is indeed much more challenging computationally in comparison to SPLPP which is already difficult to solve for many realistic real-life problems. The network has 278 stations and 277 edges. The number of potential lines is 318. The

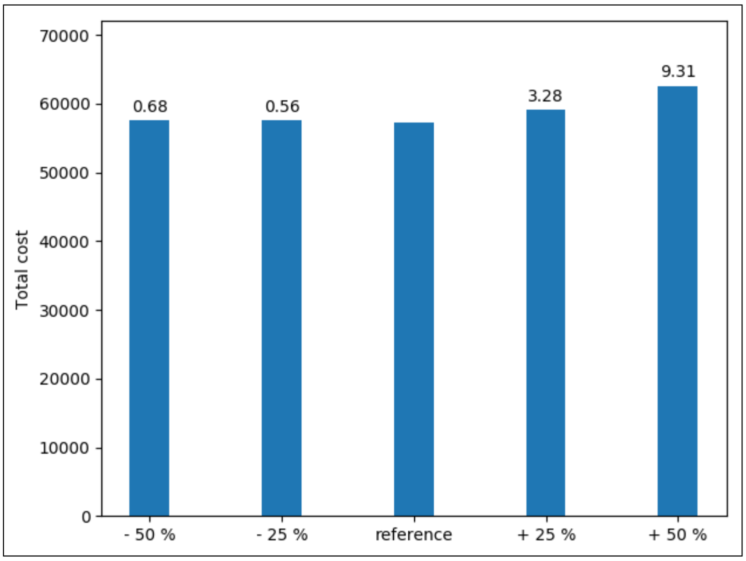


Figure 2.10 Optimal total cost values for demand data with different levels of variation for Athens instances.

Quito system was earlier studied in Torres, Torres, Borndörfer & Pfetsch (2008). Torres et al. (2008) outlines the structure of the Quito Trolébus system network. Using their cost-oriented model, they show that using both closed and open lines may reduce the total optimal cost by 50% instead of running closed lines only. It is certainly a single-period approach with static demand figures. In order to generate a mult-period demand, we use the same approach as in Section 2.2.2 for the Athens Metro and generate 19 periods of demand data covering a daily period from 5 am to midnight. We set the system parameters as $\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$.

We intend to first demonstrate the difference between SPLPP and MPLPP solutions. While solving a SPLPP for only one period takes around 10 seconds; the solver cannot find an optimal solution to the 19-period MPLPP in 24 hours. In Table 2.10, we show the results for the SPLPP solution together with MPLPP solutions obtained with the solver in 24 hours and 48 hours. Although the total cost of the solution obtained in 24 hours is significantly higher than the total cost of the solution obtained with SPLPP solutions, the solution obtained in 48 hours, with an optimality gap around 3.8%, is 10% better than SPLPP. These results clearly show that there is ample room for improvement on the solutions obtained with SPLPP solutions, it is, however, not trivial to find optimal solutions with MPLPP.

In order to test the practicality of multi-period approach and further investigate other ways to obtain multi-period solutions, at least heuristically, we investigate alternative settings. We decompose the problem with 19-hour planning horizon (5 am to midnight) into five shorter horizons: 5-9 am, 9 am-1 pm, 1-5 pm, 5-9 pm and 9 pm-12 am. The first four horizons are 4 hours long while the last is of 3 hours.

Table 2.10 A comparison between SPLPP and MPLPP solutions for the Quito Trolébus system ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

	SPLPP	MPLPP (24 hours)	MPLPP (48 hours)
Total cost	23697.02	30771.34	21504.62
Total frequency	841	1159	841
Distinct lines	19	45	25
Line usage	291	600	435
Distance traveled	20277.02	22671.30	17004.63

For each horizon, we solve a MPLPP with a time limit of 24 hours. We call this approach as the decomposed MPLPP. The results are presented in Table 2.11.

Table 2.11 A comparison between SPLPP and MPLPP solutions for the 5-period problem of the Quito Trolébus system ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

	MPLPP (48 hours)	Decomposed MPLPP
Total cost	21504.62	22903.31
Total frequency	841	844
Distinct lines	25	20
Line usage	435	301
Distance traveled	17004.63	19123.30

We first note that decomposed MPLPP is composed of five not necessarily optimal solutions, and the combined solution for the 19 hours may not be even feasible since vehicle rotations are not considered while transitioning from one horizon to the next. Despite this, the total cost of 22903.31 is larger than the feasible MPLPP solution obtained in 48 hours (21504.62). This exemplifies how the decomposed solutions can be far from a feasible multi-period solution.

While solving the MPLPP for each horizon in the decomposed approach, the time limit is 24 hours. The first 4-hour problems are not solved to optimality; the gaps are around 2-3%. The last problem with a horizon of 3 hours is solved to optimality in 24 hours since the demand figures are really low in these hours and there is ample capacity in the system. The total cost and the distance coverage of the 3-hour MPLPP solution are indeed the same as those of the combined SPLPP solutions of the three 1-hour problems. It is crucial to note that the 19-hour MPLPP cannot be solved to optimality in 48 hours and none of the 4-hour MPLPPs are also solved to optimality in 24 hours. When, the problem is large-scale and the system capacity is tight, the multi-period approach becomes extremely challenging from a computational point of view.

2.3 Multi-period LPP with Resource Transfers

A multi-period planning approach, however, convolutes the problem by bringing in essential inter-period constraints that couple the underlying single-period problems with each other (Schrage Schrage (2018)). If such constraints did not exist or could be ignored, the problem would be decomposable and each single-period problem would be solved separately. For example, the inventory balance constraints in a multi-period production planning problem relate the ending inventory level of one period to the beginning inventory level of the next period. A multi-period facility location problem has to account for the existence of facilities throughout the periods according to closing and opening decisions. In a similar but more complicated fashion, daily unit commitment problems of electricity generators are ruled by the start-up and shot-down decisions as operations associated with either of these decisions may require a couple of hours to take effect during the day. The best-known example is the simultaneous multi-period workforce and production planning problems. As such, the inter-period constraints may be necessary to ensure the availability of resources throughout the periods and their allocation among the many activities and operations associated with the process. While formulating a LPP in a multi-period setting, consideration of resource availabilities throughout the periods is inevitable. Vehicles are considered as the most critical among all resources in transportation system. Therefore, an integration of the LPP with usage of vehicles over time and allocation among lines is a realistic setting, if not the most crucial, in urban transportation systems. As mentioned earlier, a multi-period planning approach comes inherently with resource constraints and their allocation throughout periods. In this respect, we study a generalized version of MPLPP as MPLPP-VR where VR stands for vehicle rotations (transfers). As the resource units are to be allocated among the lines throughout the periods, a network flow representation (based on the discretized planning horizon) is considered. In this network representation, $G = (N, A)$ with N denoting the set of nodes and A denoting the set of arcs, as demonstrated by a schematic representation in Figure 2.11,

- node $(l, i) \in N$ represents line $l \in L$ during the period $t_i \in T$,
- a source node represents the state of resources at the beginning of the planning horizon and is identified $(0, 0) \in N$,
- a sink node represents the state of resources at the end of the planning horizon and is identified as $(0, \boldsymbol{\mathcal{T}} + 1)$, and
- an arc from node (l, i) to node (k, j) represents the flow of resource units from line l at the end of period t_i to line k at the beginning of a subsequent period t_j, i.e., $j > i$,

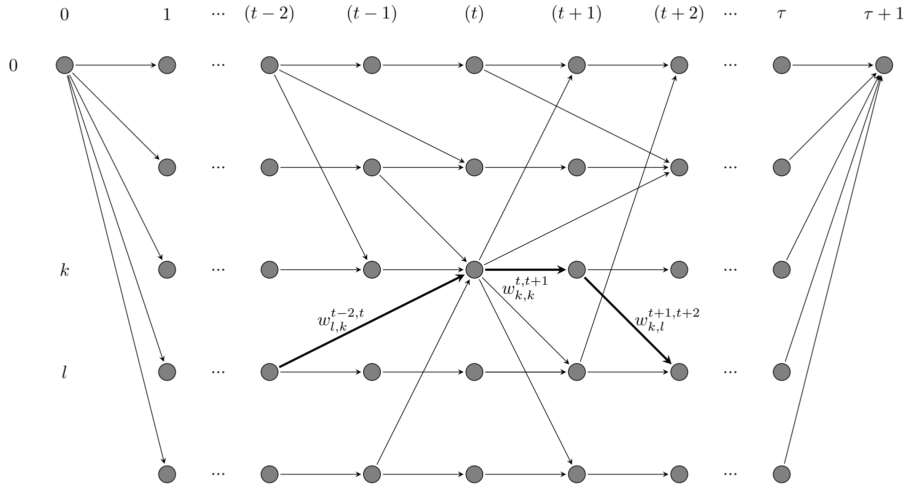


Figure 2.11 Resource transition between periods and lines

where line "0" represents the unused resource units. In Figure 2.11, $w_{lk}^{t_i t_j}$ denotes the flow from node (l,i) to node (k,j) representing the number of resource units to be transferred from line l at period t_i to line k at period t_j when $j > i$. For the purpose of computations, we exemplify the system-wide resource with the rolling stock, i.e., vehicles, which can be considered as the most critical among such resources.

In this respect, $\mathcal{T}$ denotes the length of period. We assume that the period length is the same for all periods during the planning horizon; this assumption is not restrictive. For each line $l \in L$, $\mathcal{T}_l$ represents the trip time which is calculated by considering the actual driving time, preparation time, and station time. Driving time is determined with respect to the distance covered and the vehicle speed. Fixed station time indicates the difference between arrival and departure time for loading and unloading passengers in the terminal station. Preparation time is associated with time for additional services at the start and end terminal station in each trip. $\mathcal{T}_{l,k}$ denotes the transfer time of a vehicle from the terminal station of line l to the starting station line k. $\rho_{lk} = \dfrac{\mathcal{T}_l + \mathcal{T}_{l,k}}{\mathcal{T}}$ is the number of periods required for a vehicle to transfer from line l to line k. In this respect, an integer programming problem formulation for the MPLLP-VR, becomes

$$\text{minimize} \quad \sum_{l\in L} c_l^f y_l + \sum_{l\in L}\sum_{t\in T} c_l^s v_l^t \tag{2.13}$$

$$\text{subject to} \quad \sum_{l\in L_e} \mathcal{K} v_l^t \geq d_e^t \qquad \forall e\in E, \forall t\in T, \tag{2.14}$$

$$\mathcal{W} y_l - v_l^t \geq 0 \qquad \forall l\in L, \forall t\in T, \tag{2.15}$$

$$\sum_{\substack{k\in L_0\\ t-\overline{\rho}_{kl}\geq 0}} w_{kl}^{t-\overline{\rho}_{kl},t} = v_l^t \qquad \forall l\in L, \forall t\in T, \tag{2.16}$$

$$\sum_{\substack{k\in L_0\\ t-\overline{\rho}_{kl}\geq 0}} w_{kl}^{t-\overline{\rho}_{kl},t} - \sum_{\substack{k\in L_0\\ t+\overline{\rho}_{kl}\leq \boldsymbol{\mathcal{T}}}} w_{lk}^{t,t+\overline{\rho}_{lk}} = 0 \qquad \forall l\in L_0, \forall t\in T, \tag{2.17}$$

$$\sum_{l\in L_0}\sum_{t\in T} w_{l_0 l}^{0t} = \mathcal{U}, \tag{2.18}$$

$$\sum_{l\in L_0}\sum_{t\in T} w_{l l_0}^{t|T|+1} = \mathcal{U}, \tag{2.19}$$

$$y_l \in \{0,1\} \qquad \forall l\in L, \tag{2.20}$$

$$v_l^t \in N \qquad \forall l\in L, \forall t\in T, \tag{2.21}$$

$$w_{lk}^{st} \in N \qquad \forall l,k\in L_0, \forall s\in\{0\}\cup T,$$

$$\forall t\in T\cup\{\boldsymbol{\mathcal{T}}+1\}, s<t, \tag{2.22}$$

where $L_0 = L\cup\{0\}$.

The objective function (2.13) minimizes the total cost which is composed of the fixed cost and the variable cost while the variable cost is multiplied by the service capacity in each period. Constraints (2.14) make sure that travel demand is satisfied in all periods. Coupling constraints (2.15) ensure that service is provided only on selected lines while $\mathcal{W}$ is configured to represent the inter-relations between the selection rules on lines and the limitation of the service capacities. Constraints (2.16) establish the direct relationship between w and v. Constraints (2.17)- (2.19) are the flow balance constraints controlling the transfer of vehicles from one line to another throughout the planning horizon. Finally, constraints (2.20)- (2.22) are domain constraints for all decision variables.

This section extends computational results for the presented instances for MPLPP-VR by providing exhaustive comparisons of the MPLPP and MPLPP-VR in respect to the total cost and computation times.

2.4.1 Results of MPLPP-VR for the Istanbul metrobüs system

It is not trivial to analyze the effect of considering vehicle transfers on either optimal solutions or feasibility. In the case of our baseline setting ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$), the solutions for all three daily problems are the same in MPLPP and MPLPP-VR models with respect to the total cost even when vehicle transfer constraints are considered properly as in the formulation for MPLPP-VR. In other words, the optimal solution for MPLPP is feasible for the MPLPP-VR version of the problem for the baseline setting. It is clear that the assumption of vehicle transfers in more than one period may not alter the solution and the cost when the system has ample capacity. To highlight the potential effects of considering the actual time needed to transfer vehicles, we solve problems with more limited resources when compared to the baseline setting.

In addition to the data used in MPLPP, MPLPP-VR also requires transfer times for vehicles. In order to calculate the time required for vehicle transfers, i.e., ρ_{lk} for a pair of lines l and k, we consider the trip time of a vehicle on a line and the travel time of a vehicle from the terminal station of a line to the starting station of the other:

- The trip time of a vehicle is calculated by considering preparation time and station time in addition to the actual driving time. The preparation time is concerned with additional set-up and terminating operations, respectively, before and after each trip. Station time is the duration a vehicle spends at a station stop; the difference between arrival and departure time at each station is called station time. For the sake of simplicity, we suppose that all stations have the same station time. The driving speed of a vehicle is constant for the planning horizon.
- The sum of the trip time and travel time from the ending station of a line to the starting station of the other makes up the transfer time. Then, the transfer time is divided by the length of the period to calculate the transfer time in number of periods. For a pair of lines l and k, the transfer time from l to k is, then, denoted as ρ_{lk}. For instance, if the actual travel time is 100

minutes and the period length is 60 minutes, then $\rho_{lk} = 2$ since it would take longer than one period but shorter than two periods for a vehicle to transfer from l to k. By definition, for a pair of lines l and k, $\rho_{lk} \geq 1$.

Table 2.12 shows the comparison between MPLPP and MPLPP-VR with $\mathcal{K} = 220$, $\mathcal{U} = 100$, $\mathcal{W} = 36$. The results for Weekday demand show that the total cost of MPLPP-VR is greater than that of MPLPP. This result demonstrates that all vehicles may not necessarily be available in all periods when transfers are considered. Even with the new parameter set, MPLPP and MPLPP-VR provide the same solutions for Saturday and Sunday. In order to observe the effect of MPLPP-VR with Saturday and Sunday demands, we further change the vehicle capacity, fleet size, and the maximum number of vehicles to be assigned to a line. Table 2.13 shows the results. The effect of vehicle transfer constraints is observed with Saturday demand as the total cost increases in the MPLPP-VR due to unavailability of transfers of vehicles from one period to the subsequent ones. With this new setting ($\mathcal{K} = 160$ and $\mathcal{U} = 80$), the Weekday solution is infeasible while the solutions of MPLPP and MPLPP-VR are still the same for Sunday due to ample resources in the system. The effect of vehicle transfer constraints are observable for Sunday demand only when the system resources are even more limited as $\mathcal{K} = 120$ and $\mathcal{U} = 70$ as seen in Table 2.13 again.

Table 2.12 A comparison between MPLPP and MPLPP-VR for all demand sets ($\mathcal{K} = 220$, $\mathcal{U} = 100$, $\mathcal{W} = 36$)

	Weekday		Saturday		Sunday	
	MPLPP	MPLPP-VR	MPLPP	MPLPP-VR	MPLPP	MPLPP-VR
Total cost	69790.60	70571.00	55296.80	55296.80	39790.60	39790.60
Total frequency	1103	969	937	1066	643	643
Distinct lines	6	6	5	5	5	5
Line usage	97	93	90	89	90	90
Distance traveled	68710.60	69491.00	54396.80	54396.80	38890.60	38890.60
Solution time	3.92	6.52	4.56	7.83	5.22	9.17

Table 2.13 A comparison between MPLPP and MPLPP-VR for Saturday and Sunday demand sets ($\mathcal{W} = 32$)

	Saturday ($\mathcal{K} = 160$, $\mathcal{U} = 80$)		Sunday ($\mathcal{K} = 120$, $\mathcal{U} = 70$)	
	MPLPP	MPLPP-VR	MPLPP	MPLPP-VR
Total cost	75835	76316.2	72155.80	72573.60
Total frequency	1203	1090	1067	990
Distinct lines	6	6	6	6
Line usage	91	91	99	97
Distance traveled	74755	75236.2	71075.80	71493.60
Solution time	4.21	7.28	3.31	7.26

2.4.2 Line types: closed vs. open

The concept of open lines and closed lines are well-known in public transportation planning. Line type selection may be a key design issue in line planning. In the more common version, i.e., with closed lines, the service is provided in both directions on the path from the starting station to the ending station; for each service executed in one of the directions, a service is executed in the other direction. In essence, the service frequencies are the same in both directions, and mostly vehicles go back and forth on the same path. In an open line, the service is provided only from the starting station to the ending station. A closed line corresponds to two open lines which operate on the same path in opposite directions with identical frequencies. In a line plan with closed lines only, the vehicles travel in both directions; consequently, some of the services are executed only for the sake of delivering the opposite direction rather than covering demand. Therefore, the traveled distance as well as the associated costs may increase unnecessarily. On the other hand, operating open lines may require more vehicle transfers and increase travel time for transfers resulting in a potentially more difficult-to-operate line plan. In this respect, it may be worthwhile to analyze the effect of allowable line types when the time needed to transfer vehicles between pairs of lines is an issue in the mathematical model.

We again assume that the resource related system parameters are sufficiently tight to observe the effect of transfers on MPLPP-VR solutions and set $\mathcal{K} = 220$, $\mathcal{U} = 200$, $\mathcal{W} = 36$. Table 2.14 reports the comparison among the two line types on all three daily instances. While solutions with “Closed Lines" correspond to the settings in the original baseline experiments, solutions with “Open Lines" consider the option of providing the service in only one of the directions of the original lines or running with different frequencies in opposite directions. From the cost perspective, we find out that the cost is larger when only closed lines are considered. It should also be noted that solution time is clearly larger when open lines are considered as the number of lines in the candidate set is twice as much.

Table 2.14 A comparison between line types for all demand sets ($\mathcal{K} = 220$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

	Weekdays		Saturdays		Sundays	
Type of line	Open Lines	Closed Lines	Open Lines	Closed Lines	Open Lines	Closed Lines
Total cost	55992.50	69253.40	48376.30	55296.80	37123.90	39790.60
Total frequency	2070	1197	1664	937	1132	643
Distinc lines	11	5	9	5	8	5
Line usage	180	90	161	90	141	90
Distance traveled	54012.50	68353.40	46756.30	54396.80	35683.90	38890.60
Solution time	18.70	6.05	22.46	9.86	25.41	8.66

In the case of the Istanbul Metrobüs, the demand is highly asymmetric in both time and space; this bears a huge advantage when open lines are allowed along with a multi-period planning approach. We observe this clearly in the results. Even when more lines are operated (leading to an increase in fixed costs), the optimal cost goes down by almost 20% for the Weekday which has the highest demand. This also shows that the time needed to transfer the vehicles among the starting stations of various lines may be easily compensated even when the resources are tight enough and operating open lines help to decrease the operational costs substantially. Table 2.15 shows the results of MPLPP and MPLPP-VR models for the Athens Metro. Clearly, the costs of the optimal solutions are the same. We observe that the lines in the line pool are similar with each other in terms of the transportation links they cover; for instance, some lines are extensions of others while there exist lines with the common starting (or ending) terminals. Since we consider closed lines in our computations, the travel distances between the starting terminals are very short or even 0. Therefore, considering the number of transfer periods does not cause a meaningful difference in MPLPP-VR solutions.

2.4.3 Results of MPLPP-VR for the Athens metro

The results with the Athens Metro confirm our initial findings in terms of the comparison between single-period and multi-period solutions as well as the effect of level of variation in demand. However, it also raises more questions regarding the trade-off between challenges and appeals of using more complicated models. We should also clearly note a difference between the Istanbul Metrobüs data and that of the Athens Metro: the multi-period demand data for the Istanbul Metrobüs is an exact reflection of reality with both the temporal and spatial aspects of its fluctuations and variations while the multi-period demand for the Athens Metro is artificial. The Athens demand data does not have any temporal patterns to represent peak periods, to begin with. While the mean figures for O-D pairs reflect their average popularity, generated demand figures may not necessarily reflect this due to independent randomization.

2.5 Choice of Period Length

Table 2.15 A comparison between MPLPP and MPLPP-VR ($\mathcal{K} = 250$, $\mathcal{U} = 200$, $\mathcal{W} = 36$)

	MPLPP	MPLPP-VR
Total cost	68030.58	68030.58
Total frequency	3214	3461
Distinct lines	16	16
Line usage	288	288
Distance traveled	65150583.52	65150583.52
Solution time	2667.171	2649.83

The period length in a multi-period problem is related to the temporal dimension of the O-D demand data; it specifies the time unit of the decision variables to determine the frequency of lines (through the number of vehicles of assigned to a line). The length of the period also reveals the degree of discretization of time. And, since time is indeed a continuous phenomenon, it also determines the degree of approximation. In general, the degree of approximation is higher when time is discretized in larger units. Correspondingly, shorter period length is expected to lead to more accurate and less approximate solutions in practice. Although some degree of discretization is quite necessary so that the problems can be formulated in a discrete space and solutions shall be interpreted easily, its effect on the solution of the problem in terms of resource usage may not be as trivial as the accuracy. In this respect, we aim to investigate how both the accuracy and effectiveness of the solution change when alternative period lengths are used. For the original Metrobüs demand data, the period length is one hour, i.e., 60 minutes. We now consider three scenarios for the length of the period: 60 minutes, 30 minutes and 15 minutes. The original demand data is transferred to shorter periods by allocating the demand of a longer period to a set of shorter ones by interpolating and smoothing out the demand according to the demand amount in previous and subsequent periods of the original longer period version. Therefore, the interpolator ensures that monotonicity is maintained in the interpolated demand, even when the demand is not smooth. Generally speaking, the demand is interpolated based on the period length as the period length is shortened. Regardless of the length of the period, the total demand remains unchanged. The transfer matrix for vehicle rotation is also updated for alternative period lengths.

Tables 2.16 - 2.18 report the results with MPLPP-VR formulation for the Metrobüs instance. Looking at the solution metrics closely, we observe that the total cost for the Weekday (see Table 2.16) first decreases when the period length goes from 60 to 30 minutes; but, then it increases again when the period length goes from 30 to 15 minutes. With the same parameter setting ($\mathcal{K} = 220$, $\mathcal{U} = 100$), however, for Saturday and Sunday, the total cost increases as the period length is shortened. We again check the results when system resources are tighter at a level for which the Weekday solution is already infeasible. Table 2.17 shows solutions for Saturday. Changing the period length form 60 to 30 minutes leads to a decrease in the total cost with $\mathcal{K} = 160$ and $\mathcal{U} = 80$; cost increases when the period length goes further down to 15 minutes. When $\mathcal{K} = 120$ and $\mathcal{U} = 70$, the total cost for Sunday (see Table 2.18) follows the same trend as that for Weekday with $\mathcal{K} = 160$ and $\mathcal{U} = 100$ and Saturday with $\mathcal{K} = 160$ and $\mathcal{U} = 80$. As a matter of fact, we make the following observations:

- It is probable that shortening the period length may decrease the total cost and provide even a better solution while increasing the accuracy.
- While shorter period lengths contribute to accuracy of the solutions by satisfying the demand in a timely manner since frequencies are arranged for shorter time periods, it may lead to inefficient use of resources due to discrete nature of resource capacities and shorter periods (leading to an increase in the number of periods) can lead to more slack in resource capacities.

These results clearly show that the period length is not a trivial decision.

Table 2.16 Comparison among alternative period lengths with Weekday demand ($\mathcal{K} = 220$, $\mathcal{U} = 100$)

Period length (minutes)	60	30	15
$\mathcal{W}$	36	18	9
Total cost	70,571.00	70,386.40	71,916.00
Total frequency	969	1,210	1,230
Distinct lines	6	5	5
Line usage	93	180	359
Distance traveled	69,491.00	69,486.40	71,016.00
Solution time	7.36	22.83	77.63

Table 2.17 Comparison among alternative period lengths with Saturday demand

	($\mathcal{K} = 220$, $\mathcal{U} = 100$)			($\mathcal{K} = 160$, $\mathcal{U} = 80$)		
Period length (minutes)	60	30	15	60	30	15
$\mathcal{W}$	36	18	9	32	16	8
Total cost	55296.80	56239.00	58134.00	76316.20	76196.40	78245.60
Total frequency	1066	1085	1,122	1090	1289	1321
Distinct lines	5	5	5	6	5	5
Line usage	89	179	359	91	179	356
Distance traveled	54396.80	55339.00	57234.00	75236.20	75296.40	77345.60
Solution time	8.50	26.58	78.68	9.74	18.96	72.08

The effect of period length is not only observed as an increase or a decrease in the total cost. It may also alter the feasibility of the problem setting. In Table 2.19, we show the results with a different set of system parameters where $\mathcal{K} = 150$ and $\mathcal{U} = 100$ for the Weekday demand data set. When the period length is 60 minutes,

Table 2.18 Comparison among alternative period lengths with Sunday demand

	($\mathcal{K} = 220, \mathcal{U} = 100$)			($\mathcal{K} = 120, \mathcal{U} = 70$)		
Period length (minutes)	60	30	15	60	30	15
$\mathcal{W}$	36	18	9	32	16	8
Total cost	39790.60	40809.80	42521.00	72573.60	72567.80	74210.60
Total frequency	643	654	672	990	1375	1404
Distinct lines	5	5	5	6	5	5
Line usage	90	176	337	97	180	360
Distance traveled	38890.60	39909.80	41621.00	71493.60	71667.80	73311.00
Solution time	10.36	38.74	121.24	7.80	21.19	108.01

the problem does not even have a feasible solution while the solutions for shorter period lengths are feasible.

Table 2.19 Effect of length of period on the solution for Weekday instance with $\mathcal{K} = 150$ and $\mathcal{U} = 100$

Period length (minutes)	60	30	15
$\mathcal{W}$	36	18	9
Total cost		102430.60	104287.60
Total frequency		1699	1709
Distinct lines	inf	7	7
Line usage		197	381
Distance traveled		101170.60	103027.60
Solution time		17.94	68.29

In Tables 2.16 - 2.19, we also observe that the solution time for the solver increases significantly when the period length is shortened, which increases the problem size due to both number of variables and constraints.

We investigate the intriguing effect of the period length also for the Athens Metro data and repeat the same setting. The results are presented in Table 2.20. The intriguing effect is only confirmed: the total cost of optimal solution increases directly for Athens data when the period length goes from 60 minutes to 30 minutes, i.e. the need for more accuracy naturally increases the cost unlike the Metrobüs. Meanwhile, we also face with an expected computational challenge when the period length further goes to 15 minutes and the size of the problem gets larger. In 24 hours, the solver cannot even create the model to solve problem.

Table 2.20 Alternative period lengths for Athens instance with $\mathcal{K} = 250$, $\mathcal{U} = 200$

Period length (minutes)	60	30	15
$\mathcal{W}$	36	18	9
Total cost	68030.583	70080.89	-
Total frequency	3461	3305	-
Distinct lines	16	16	-
Line usage	288	576	-
Distance traveled	65150583.52	67200893.82	-
Solution time	2649.83	27560.94	-

It turns out that the length of the period may be an intricate choice to make; its effect firstly manifests itself as a trade-off between accuracy and effectiveness of the solutions. On the one hand, the solutions are likely to suffer from accuracy, leading to both unutilized capacity and unserved transport demand when the periods are too long. On the other, shorter period lengths are expected to help with accuracy at the expense of increasing the use of resources inefficiently. In addition, shorter period lengths also increase the computational challenges in practice. The trade-off between the accuracy and the effectiveness of the solutions coupled with the computational challenges due to the size of the problem make the choice of period length a subject of optimization. We also note that a viable range for period length should be considered case by case. While the Metrobüs seems to be an ideal example to investigate the effect of period length on the solution, we could only try longer period lengths with the Quito Trolébus in order to solve the problem to optimality.

2.6 Concluding Remarks

We present a multi-period planning approach for the well-known LPP; our approach is motivated by the drawbacks of traditional static line planning approaches for not being able to consider the dynamism of the demand. In practice, the traditional approaches may still work for systems with moderate demand load and where target service levels are already achieved with more than sufficient resources. For such systems, it is usually trivial to identify peak loads with respect to time and space and mostly as well as directions on the network. However, for overly-crowded systems for which many examples can be found as BRTs in different cities, unwanted passenger waiting times at stations resulting in longer travel times and lower service levels shall be handled if the changes in travel demand in time are considered explicitly. The new approach proposes consideration of longer planning horizons which are divided into periods of manageable length in terms of planning and coordination of both services and resources throughout the periods in the planning horizon.

We characterize the demand as a function of time, first; this helps us develop a continuous-time LPP for the first time in the literature. Then, for practical purposes, we develop an integer programming problem formulation for the MPLPP through discretization of the continuous planning horizon. In our computational study, we first work with this problem and show that

- both solutions and resulting costs (represented by the objective functions) are improved significantly when a multi-period approach is employed as an alternative to combined solutions of traditional single-period line plan solutions;
- higher variation in demand benefits even more from a multi-period approach as higher fluctuations of demand in time leads to higher system costs even when the total demand does not change.

As a matter of fact, we are able to experimentally show that a multi-period approach shall outperform a traditional single-period approach under various circumstances.

As easily and clearly observed from many examples in the literature and practice, decision-making and optimization with multi-period planning approaches naturally involve resource planning. Indeed, planning of resources is mostly what couples the time periods in a typical problem formulation. In this respect, we develop a generalization of the first MPLPP formulation by integrating resource allocation and transfer constraints. Computations with this problem show that solutions may change significantly when resource constraints are involved and tight. Therefore, it is necessary to employ an approach where resource transfers are also included in order to obtain realistic and implementable solutions. The integration of resource constraints may require even more complex models: including deadheading costs, accounting for the utilization of infrastructure by the deadheading vehicles, and vehicles of different types.

We also observe that the computational challenges of well-known LPP formulations are naturally inherited by the multi-period approach. For large-scale instances, the multi-period version where the period length is an hour and the planning horizon is a day cannot be solved to optimality even in 48 hours while the underlying single-period problem with fixed costs may be solved in reasonable time despite its NP-hardness. The size of a problem depends on the size of the network, the number of possible lines and also the number of periods in the planning horizon whose determinant is the length of the period.

Last but not the least, our computations show that choosing the period length may be an intricate decision that is justified by a trade-off between accuracy of the solutions and efficiency of resource planning as well as the computational effort to solve the problem. In contrast to general understanding of time discretization, shorter time periods may not necessarily lead to better solutions. The choice of the period length constitutes a paradox: shorter periods improve the accuracy of the solution at the expense of increasing the the costs and degrading the capacity utilization. A careful analysis of the change in demand may also necessitate non-identical period lengths in a planning horizon, which may be a remedy to counter-effect the trade off between accuracy and effectiveness. However, it clearly requires a detailed understanding of the demand pattern.

3. A LOCAL BRANCHING ALGORITHM FOR SOLVING MPLPP-VR

In Section 2, we consider a multi-period approach for the LPP. For the MPLPP, the planning horizon is discretized into a sequence of time periods with a predetermined length; the demand information is provided for every single period. To overcome the shortcomings of traditional static line planning approaches, we consider the rotation of vehicles between pairs of lines. We show that the MPLPP-VR formulation becomes more complicated due to the constraints coupling the periods. We observe that a multi-period approach is 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 resort to a local branching algorithm that can be scaled to solve multi-period line planning problems with vehicle transfers even for a very-large PTN.

In the literature, various advanced optimization techniques are applied to solve the LPPs. As mentioned earlier, Claessens et al. (1998), Bussieck (1998), and Goossens (2004) are mostly focusing on the Branch and Bound algorithm and improve their approach by defining strong valid inequalities. On the other hand, some studies employ different optimization techniques to solve the problem in a short computational time. Schöbel & Scholl (2006) present a Dantzig-Wolfe decomposition approach to solve the LP-relaxation since the suggested model has a block diagonal structure. Borndörfer et al. (2007) propose a column-generation approach such that the pricing problem is polynomially solvable. Bull, Rezanova, Lusby & Larsen (2016) present a problem formulation by using multi-commodity flows for the Copenhagen rail system such that the travel time is minimized. They propose an LP-based heuristic approach to solve the problem. Results show that the algorithm finds a high-quality feasible solution but not necessarily an optimal solution.

Following these advanced optimization techniques, we exploit a local branching algorithm to solve the MPLPP-VR. Local branching is an iterative method which may provide a high-quality incumbent solution within an acceptable computational time (Fischetti & Lodi, 2003). 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 (TT) limit or the maximum number of diversifications, are reached.

Following Fischetti & Lodi (2003), Fischetti, Polo & Scantamburlo (2004) propose an integrated approach to hybridize a local branching algorithm and a variable neighborhood search algorithm. They apply the proposed heuristic method to specific MIP problems where binary decision variables are separated into two levels. They solve easier problems in the second level by fixing binary variables beforehand. They show that their approach provides satisfactory results in the telecommunications network design problem. Rodríguez-Martín & Salazar-González (2010) solve the capacitated fixed-charge network design problem with the local branching algorithm. The computational results demonstrate that the proposed algorithm finds better solutions than the other heuristic approaches. Legato & Trunfio (2014) employ an integrated approach by laying the local branching within a refined branch-and-bound (B& B) algorithm to solve the crane scheduling problem. They demonstrate the effectiveness by presenting extensive comparisons against the efficient algorithms in the literature. Smet, Wauters, Mihaylov & Berghe (2014) propose a two-phase hybridization metaheuristic method to solve a shift minimisation personnel task scheduling problem. In their approach, a feasible solution is generated in the first phase by a constructive heuristic; in the second phase, they resort to the local branching algorithm to improve the feasible solution.

Based on such satisfactory outcomes with various combinatorial optimization and integer programming problems in the literature, we aim to study if local branching outperforms commercial solvers when applied to the MPLPP-VR. We compare local branching results with those currently obtained in Section 2. We also implement a Lagrangian Relaxation (LR) algorithm to provide a lower bound on the optimal solution of the problem.

The remainder of this chapter is organized as follows. In Section 3.1, we study the local branching algorithm for the MPLPP-VR, while, in Section 3.2, we present computational results for the local branching algorithm. Finally, in Section 3.3, we provide concluding remarks.

3.1 Solution Algorithm for MPLPP-VR

The local branching algorithm is introduced as an exact method, in principle, to enhance the performance of the heuristic algorithms that are used in the MIP solvers (Fischetti & Lodi, 2003). Given a feasible integer solution, the local branching algorithm defines a neighborhood by adding a cut to the original problem. Each cut results in a restricted sub-problem that includes the solutions within a distance less than or equal to a predefined parameter $k > 0$ from the initial feasible solution, i.e. the values of at least k decision variables are altered. Solving the restricted sub-problem makes the search procedure easier to find a better incumbent solution within a given computation time limit. When the algorithm finds a better feasible solution, we add a new branching cut to the current sub-problem; otherwise, it exploits a diversification mechanism to move to another point in the solution space whenever the local branching algorithm verifies that there are no improving feasible solutions. A diversification mechanism enables the algorithm to enlarge or compress the search region. With this iterative approach, the MIP solver explores a pool of the smaller sub-problems to find an optimal solution. The algorithm terminates with predefined stopping criteria are met with respect to the TT limit or the maximum number of diversifications are reached.

3.1.1 Local branching algorithm

Fischetti & Lodi (2003) describe the local branching algorithm considering a general combinatorial optimization problem as

$$
\begin{aligned}
(\mathcal{P}) \quad \text{minimize} \quad & c^T y \\
\text{subject to} \quad & Ay \geq b, \\
& y_j \geq 0 && \forall j \in \mathcal{I}, \\
& y_j \in \{0,1\} && \forall j \in \mathcal{B},
\end{aligned}
$$

where set $\mathcal{I}$ and $\mathcal{B}$ correspond to the index of the integer and binary variables, respectively. The local branching is initiated by a feasible solution y^0 as a reference solution. Given a feasible reference solution of $(\mathcal{P})$, binary variables are divided into two sets $\bar{\mathcal{S}}_0 = \{j \in \mathcal{B} : y_j^0 = 1\}$ and $\mathcal{S}_0 = \mathcal{B} \setminus \bar{\mathcal{S}}$. To define the neighborhood $N(y^0, k)$, the local branching cut, i.e., the left-branching cut, is added to the original formulation of the problem $(\mathcal{P})$ as

$$
\delta(y^0) = \sum_{j \in \bar{\mathcal{S}}_0} (1 - y_j^0) + \sum_{j \in \mathcal{S}_0} y_j^0 \leq k. \tag{3.1}
$$

If a new solution is obtained, inequality (3.1) is replaced with a new local branching cut, i.e., the right-branching cut, as

$$
\delta(y^0) = \sum_{j \in \bar{\mathcal{S}}_0} (1 - y_j^0) + \sum_{j \in \mathcal{S}_0} y_j^0 \geq k + 1. \tag{3.2}
$$

The new cut guarantees that the already explored feasible solution space will not be explored again.

The local branching is an iterative algorithm. At each iteration, it either generates a left-branching cut and a right-branching cut or uses the diversification mechanism to find a new solution. According to Fischetti & Lodi (2003), after solving the restricted sub-problem with the left-branching cut

$$
\delta(y^{i-1}) = \sum_{j \in \bar{\mathcal{S}}_{i-1}} (1 - y_j^{i-1}) + \sum_{j \in \mathcal{S}_{i-1}} y_j^{i-1} \leq k, \tag{3.3}
$$

at iteration i within a given TT limit, one of the following cases might arise.

Case 1 *y^i is an optimal solution. According to the new reference solution, y^i, $\bar{\mathcal{S}}_i$ and $\mathcal{S}_i$ are generated. The last left-branching cut* (3.3) *is replaced with the right-*

branching cut,

$$\delta(y^{i-1}) = \sum_{j \in \bar{\mathcal{S}}_{i-1}} (1 - y_j^{i-1}) + \sum_{j \in \mathcal{S}_{i-1}} y_j^{i-1} \geq k + 1. \tag{3.4}$$

A new left-branching cut is added to the restricted sub-problem by using the solution y^i. *Figure 3.1 depicts the schematic view of Case 1.*

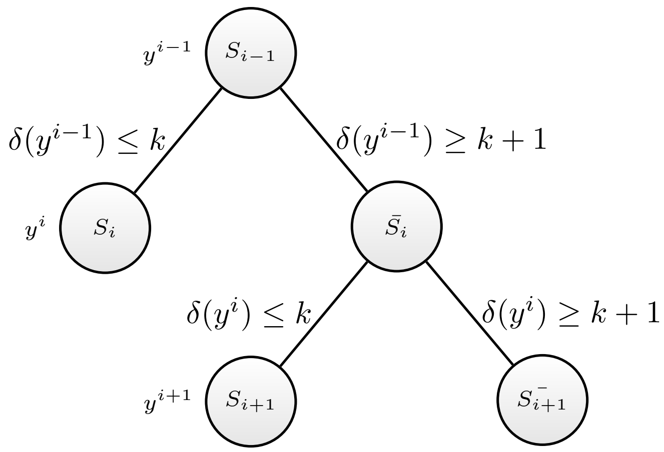


Figure 3.1 Local branching: Case 1

Case 2 *The restricted sub-problem is infeasible. The feasible region of the current restricted sub-problem is enlarged by a diversification mechanism. To do this, the right-hand side of the last left-branching cut* (3.3) *is increased by* $\lceil k/2 \rceil$. *Figure 3.2 displays the schematic idea of Case 2.*

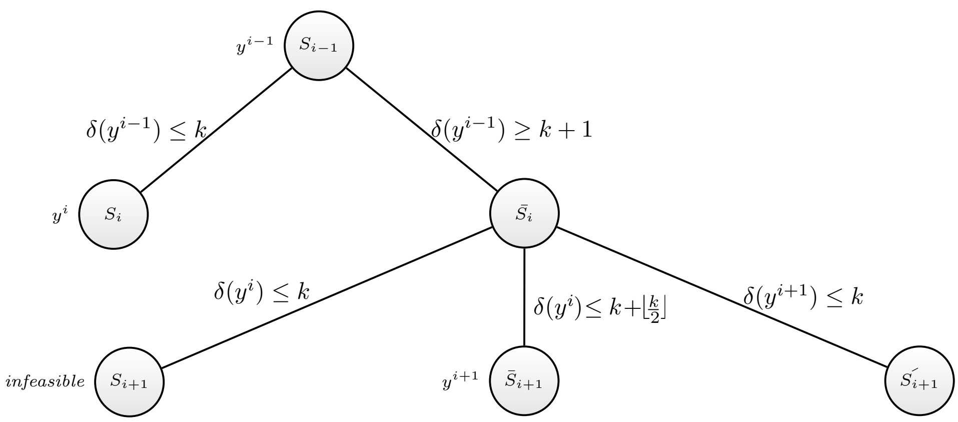


Figure 3.2 Local branching: Case 2

Case 3 *The node time (NT) limit is reached and the feasible solution, say* y^i, *is better than the best reference solution while it is not an optimal solution. Since the solution space of the current restricted sub-problem has not been explored thoroughly,* *the most recent left-branching cut* (3.3) *is deleted, based on the new reference solution, y^i, $\bar{\mathcal{S}}_i$, and $\mathcal{S}_i$ are generated, and a new left-branching cut is added to the restricted sub-problem. Figure 3.3 demonstrates the schematic view of Case 3.*

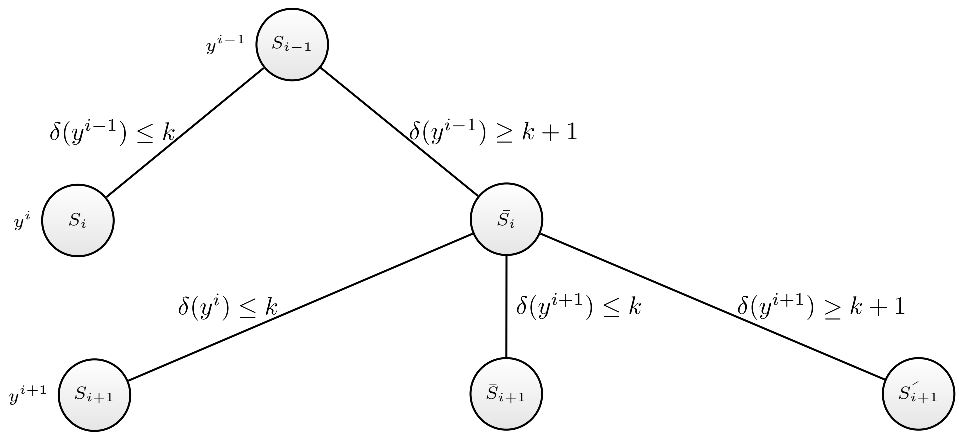


Figure 3.3 Local branching: Case 3

Case 4 *The NT limit is reached and there is no improvement in the solution of the current restricted sub-problem. The most recent left-branching cut* (3.3) *is deleted and the parameter k is reduced by $\lceil k/2 \rceil$. A new left-branching cut is added to the restricted sub-problem by using the reference solution y^{i-1} and the updated k. If no improvement is found in the updated restricted sub-problem, the diversification mechanism is applied to enlarge the neighborhood by adding $\lceil k/2 \rceil$ to the right-hand side of the left-branching, to jump to another point in the solution space. The last left-branching cut is replaced by a Tabu cut*

$$\sum_{j \in \bar{\mathcal{S}}_{i-1}} (1 - y_j^{i-1}) + \sum_{j \in \mathcal{S}_{i-1}} y_j^{i-1} \geq 1. \tag{3.5}$$

Figure 3.4 presents the schematic view of case 4.

The proposed approach is outlined in Algorithm 1. *bestUB* and *bestSolution* respectively denote the best upper bound and best solution until the current iteration. $SolCount$ is a conditional parameter that determines which alternate solution is retrieved. If $SolCount$ is $true$, the first incumbent solution is selected; otherwise, the solver chooses the last incumbent solution found during NT. max_{dv} is the maximum number of diversifications and dv shows the number of diversifications until the current iteration.

3.1.2 Lagrangian relaxation

Algorithm 1 Local branching

$UB \leftarrow +\infty$; $bestUB \leftarrow +\infty$; $rhs \leftarrow k$; $itr \leftarrow 0$;
$diversity \leftarrow false$; $dv \leftarrow 0$; $SolCount \leftarrow false$;
while $elapsed - time \leq TT \quad and \quad dv < max_{dv}$ **do**
$\bar{y} \leftarrow y^{itr}$;
add left-branching cut (3.3);
SOLVE(NT,UB, $SolCount$, $\bar{x}$);
if Case 1 **then**
if Objvalue $< bestUB$ **then**
$bestUB \leftarrow$ Objvalue; $bestSolution \leftarrow y^{itr}$;
end if
replace left-branching cut (3.3) with right-branching cut (3.4);
$diversity \leftarrow false$; $SolCount \leftarrow false$; $rhs \leftarrow$ k; $UB \leftarrow$ Objvalue;
end if
if Case 2 **then**
if $rhs > +\infty$ **then**
replace left-branching cut (3.3) with right-branching cut (3.4);
end if
if $diversify$ **then**
$UB \leftarrow +\infty$; $dv \leftarrow dv + 1$; $SolCount \leftarrow$ true;
end if
replace left-branching cut (3.3) with right-branching cut (3.4);
diversity $\leftarrow$ true; rhs $\leftarrow rhs + \lceil k/2 \rceil$;
end if
if Case 3 **then**
if Objvalue $< bestUB$ **then**
$bestUB \leftarrow$ Objvalue; $bestSolution \leftarrow y^{itr}$;
end if
delete left-branching cut (3.3);
$diversity \leftarrow false$; $SolCount \leftarrow false$; $rhs \leftarrow$ k; $UB \leftarrow$ Objvalue;
end if
if Case 4 **then**
if $diversify$ **then**
replace left-branching cut (3.3) with Tabu cut; (3.5);
$UB \leftarrow +\infty$; $dv \leftarrow dv + 1$; $SolCount \leftarrow true$; $rhs + \lceil k/2 \rceil$
else
delete last left-branching cut (3.3);
$rhs - \lceil k/2 \rceil$;
end if
end if
end while

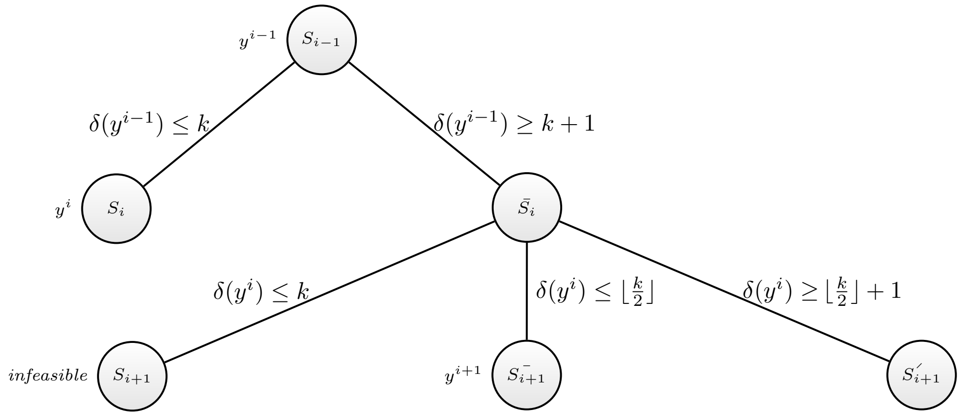


Figure 3.4 Local branching: Case 4

We also develop a lagrangian relaxation scheme in order to provide a lower bound. Let $\bar{\lambda} = \{\lambda_l^t \in \mathbb{R}_{\ge 0} : \forall l \in L, \forall t \in T\}$ represent the set of Lagrangian multipliers associated with constraints (2.15). Then the relaxed MPLPP-VR (MPLPP-VR$_\lambda$) can be written as

$$\text{(3.6)} \qquad \text{minimize} \quad \sum_{l \in L} c_l^f y_l + \sum_{l \in L} \sum_{t \in T} c_l^s v_l^t + \lambda_l^t (v_l^t - \mathcal{W} y_l)$$

$$\text{subject to} \quad (2.14), (2.16) - (2.22).$$

Relaxing constraints (2.15) make the remaining problem easier to solve. To update the Lagrangian multipliers, we use the sub-gradient method which adjusts the Lagrangian multipliers by reducing the constraint violation. Given an initial vector $\bar{\lambda}^0$, the MPLPP-VR$_\lambda$ is solved. Let us define $\hat{y}^k = \{(\hat{y}_l)^k : l \in L\}$ as the set of corresponding optimal solution at iteration k. The Lagrangian multipliers are updated by the rule

$$\text{(3.7)} \qquad (\lambda_l^t)^{k+1} = (\lambda_l^t)^k + S_k (v_l^t - \mathcal{W}(\hat{y}_l)^k),$$

where S_k is a positive scalar step size. In practice, the step size is determined by

$$\text{(3.8)} \qquad S_k = \frac{\mu_k (Z_{UB} - Z_{LB})}{\left\| v_l^t - \mathcal{W}(\hat{y}_l)^k \right\|^2},$$

where Z_{LB} is the objective function value of the MPLPP-VR$_\lambda$. $\mu_k \in (0,2)$ is a scalar and halved whenever Z_{LB} does not improve after some fixed number of iterations. Z_{UB} is obtained by applying the local branching on the MPLPP-VR (see section 3.1.1).

3.2 Computational Experiments

We study the computational efficiency of the local branching algorithm for the PTN instances we use in Chapter 2. First, we apply the local branching algorithm to solve the MPLPP-VR and compare solutions with the solutions provided by Gurobi. Then, we investigate the convergence behavior of the proposed algorithm within a given TT limit. Finally, we analyze how the solution quality changes by altering the parameters of the local branching algorithm. We also create a new instance based on the Quito system with 122 potential lines (Quito-122 hereafter), instead of 318 original lines. The edges and the stations are the same as the Quito system with 318 lines (Quito-318 hereafter).

All computational experiments are executed on a computer with Intel Core(TM) i7-4770 CPU v2 3.40 GHz CPU and 16 GB RAM, using Gurobi Optimizer 9.0.1 as the IP solver with Python 3.7.4.

We consider $\mathcal{K}$=250, $\mathcal{U}$=200, and $\mathcal{W}$=36. The base parameter setting for the local branching algorithm parameters are defined as follows.

- k is equal to 3, 15, 30, and 80 for the Metrobüs, Athens, Quito-122, and Quito-318, respectively. These values are 25% of the size of the line pool.
- The NT limit for the Metrobüs system, Athens system, Quito-122 and Quito-318 are 15, 90, 1800, and 5400 seconds, respectively. The node times increase in proportion to the size of the line pool. Correspondingly, the TT limits are ten times the NT limit, which are 150, 900, 18000, and 54000 seconds.
- As the secondary termination criteria, the maximum number of diversification is considered 5 for all instances.

We first assess the computational performance and the quality of the solutions obtained by the local branching algorithm. Table 3.1 presents a comparison for the local branching algorithm against Gurobi. The first column reports the total cost, the sum of the fixed and operating costs, the second column represents the CPU times in seconds. Results indicate that the local branching algorithm is fairly efficient. According to Table 3.1, the local branching algorithm provides the optimal solution for the Metrobüs system for all instances with different daily demands (i.e., Weekday, Saturday, and Sunday) and the Athens system. The algorithm also outperforms Gurobi in terms of computational time. The solution of the Athens system indicates that the local branching algorithm improves the CPU time by approximately 40 percent. The algorithm performed well on the Quito-122 system, yielding a high-quality solution within approximately three times less computation time. The Quito-318 system is much more challenging than the Quito-122 system. The results for the Quito-318 system show that the algorithm obtains a feasible solution almost as good as the one obtained with Gurobi within approximately less than 67% of the time used by Gurobi. Initial results exhibit reasonable performance both in terms of the computational time and obtaining a high-quality solution fast enough.

Table 3.1 Results obtained by Gurobi and the local branching algorithm ($\mathcal{K}$=250, $\mathcal{U}$=200, $\mathcal{W}$=36).

Instance	Gurobi		Local branching		Lower bound	Gap(%)
	Cost	Time	Cost	Time		
Istanbul-Weekday	61334.60	< 1	61334.60	< 1	60965.97	0.60
Istanbul-Saturday	48807.00	< 1	48807.00	< 1	48145.82	1.35
Istanbul-Sunday	35377.80	< 1	35377.80	< 1	34629.00	2.11
Athens	68030.58	2071	68030.58	1284	65941.19	3.07
Quito-122	21690.17	59135	21691.65	18000	16322.35	24.75
Quito-318	21611.21	86400	21618.23	54000	15594.70	27.86

We further use the Lagrangian relaxation to obtain a lower bound. we determine the gap as $(UB - LB)/LB$ where UB and LB represent the bounds found by the local branching and the LR, respectively. The algorithm starts with initial Lagrangian multipliers set to 0. The two last columns of Table 3.1 show he lower bound obtained by the Lagrangian relaxation within the time limit and the gap, respectively. According to the results in Tables 3.1, the obtained gaps also certify that the local branching algorithm can find a solution very close to the optimal one in a reasonable time for the small and medium instances.

We also explore the convergence behavior of the local branching algorithm within the TT limit. Figure 3.5 depicts the four evolution diagrams, one for each instance. They show how the solutions change as the computation proceeds during the local branching algorithm. Since for the Metrobüs system, these changes are similar in all three demand data (i.e., Weekday, Saturday, and Sunday), we only illustrate the total cost changing scheme for Weekday demand data. For the sake of comparison, the diagrams present the total cost changes with respect to the CPU run time. According to the figures, in all instances, the convergence of the local branching is quite satisfactory. They show that the local branching algorithm finds a near-optimal solution quickly in early iterations and is gradually stable in late iterations.

The local branching algorithm performance is highly dependent on both k and NT limit. Defining an effective choice for parameters is an important decision in the

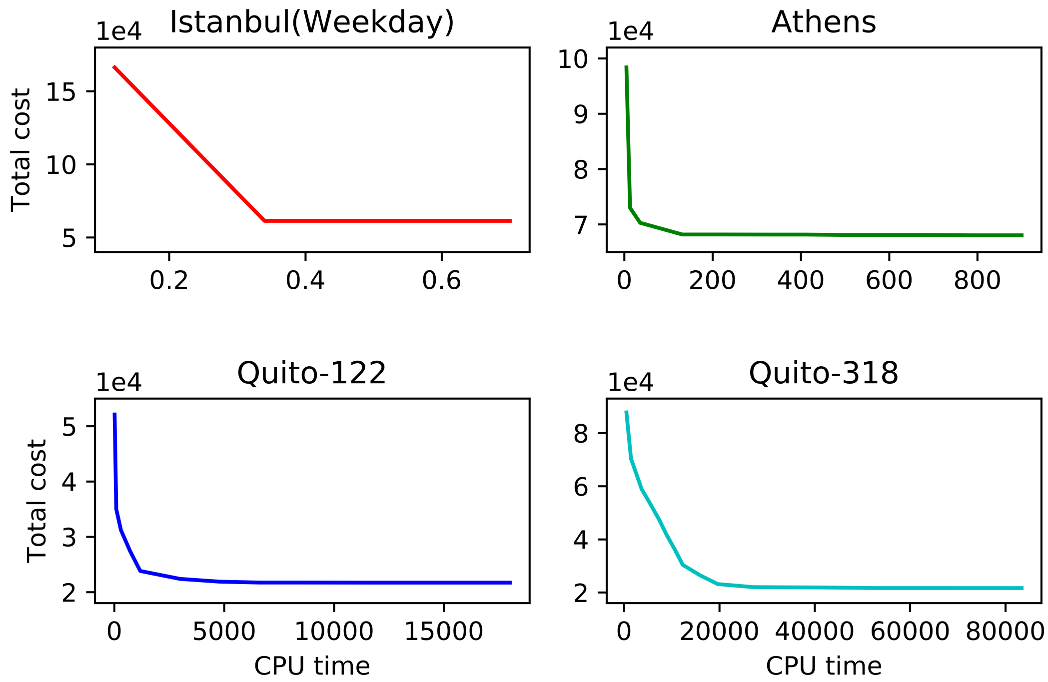


Figure 3.5 Performance of the local branching algorithm within the TT limit on the four instances.

algorithm (Fischetti & Lodi, 2003). Accordingly, we proceed to investigate the impact of both k and the NT limit on the quality of the solution and computational time of the local branching algorithm. We provide details of our analyses in Table 3.2 and Table 3.3. For the Metrobüs system, since the computation time is shorter than one second, there is no noticeable difference in the CPU run time with alternative values of either k or the NT limit. Therefore, we exclude the Metrobüs system from further investigation. In Table 3.2 we fix the NT limit for each instance and experiment with alternative values of k. The first column illustrates the alternative values for k; the total cost and the diversification within the TT limit are reported in the second and third columns, respectively.

For the Athens system, the algorithm finds the optimal solution when k equals 15 or 20; the algorithm still finds good-quality feasible solutions with other values of k. Results also show that the most reasonable CPU times are obtained when k equals 15 (initial setting) for the Athens system. For the Quito-122 instance, we observe that the algorithm reaches the TT limit for all k values. When k decreases from 30 to 20, the algorithm yields a better optimality gap. For the Quito-318 instance, there is no prominent difference in the quality of the solutions with different k values. The results show that changing k has no significant effect on the quality of the solutions. This could be to the already limited performance on this particular instance.

Table 3.2 Results for Athens, Quito-122 and Quito-318 instances with different k.

Instance	k	Solution time	Total cost	Diversification
Athens (NT =90 s, TT=900 s)	10	900	68053.86	0
	15	698	68030.58	0
	20	877	68030.58	1
	25	900	68053.86	1
	30	900	68048.67	1
	35	900	68048.67	1
	40	900	68122.77	2
Quito-122 (NT =1800 s, TT=18000 s)	20	18000	21723.87	0
	30	18000	21708.75	0
	40	18000	21753.07	1
	50	18000	21741.15	0
	60	18000	21754.11	1
	70	18000	21749.79	0
	80	18000	21753.07	1
Quito-318 (NT =5400 s, TT= 54000 s)	60	54000	21643.67	1
	70	54000	21638.53	1
	80	54000	21618.23	1
	90	54000	21625.99	1
	100	54000	21662.03	1
	110	54000	21621.89	0
	120	54000	21632.33	1

Table 3.3 Results for Athens, Quito-122 and Quito-318 instances with different NT limit.

Instance	NT	Solution time	Total cost	Diversification
Athens (k=15, TT=900 s)	30	900	68052.38	3
	45	900	68052.38	1
	60	741	68030.58	0
	90	698	68030.58	0
	120	900	68099.01	2
	150	718	68030.58	0
	180	900	68048.67	1
Quito-122 (k =30, TT=18000 s)	600	18000	21708.75	2
	1200	18000	21707.85	0
	1800	18000	21708.75	0
	2400	18000	21738.55	1
	3000	18000	21738.55	1
	3600	18000	21738.55	0
	4200	18000	21706.51	0
Quito-318 (k =80, TT=18000 s)	1800	54000	21899.22	2
	3600	54000	21631.25	1
	5400	54000	21618.23	1
	7200	54000	21590.21	0
	9000	54000	21603.17	0
	10800	54000	21613.85	0
	12600	54000	21622.49	0

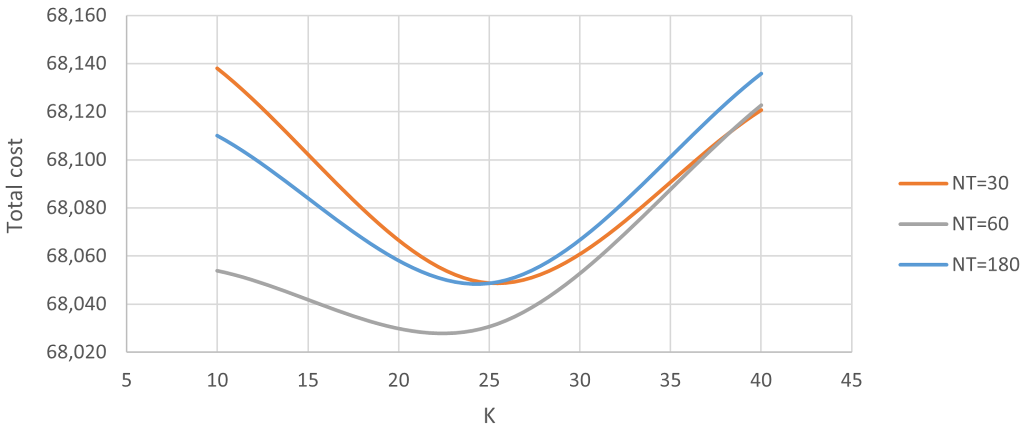


Figure 3.6 Comparison of the total cost of Athens over various k and the NT limit.

In Table 3.3, we fix k and change NT. In these experiments, the TT limit is the same as the base setting in Table 3.1. For the Athens system, the results show that the optimal solution is obtained with three NT limits, i.e., 60, 90, and 150 seconds. Results for the Quito-122 instance show that with smaller value for NT limits, the algorithm obtains high-quality solutions. Similarly, the results for the Quito-318 instance yield similar observations. According to these results, it would be fair to claim that the NT limit may not significantly impact the performance of the local branching algorithm.

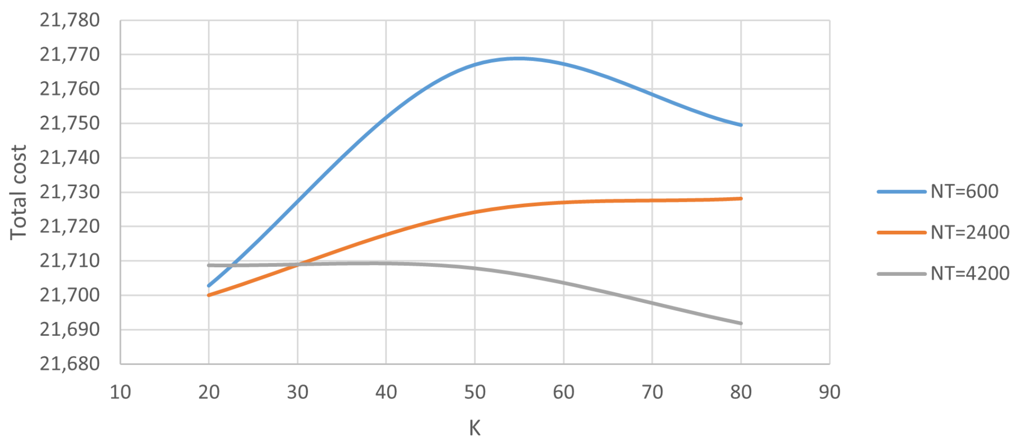


Figure 3.7 Comparison of the total cost of Quito- 122 over various k and the NT limit.

Consistent with the results in Table 3.2 and Table 3.3, Figures 3.6-3.8 present the impact of k of on the quality of the solution with respect to the three alternative values for NT limits. Figure 3.6 shows that increasing k to 25 leads to better quality solutions with the Athens system for various values of NT limits scenarios, . However, larger k values lead to lower-quality solutions with higher total costs.

In the Quito-122 instance, Figure 3.7, when the NT limit is larger, choosing larger

k values is more effective. In contrast, when the NT limit decreases, the algorithm fails in finding a better feasible solution with larger k.

Contrary to our observations for the Athens system, Figure 3.8 shows that the solutions become worse when k is increased to 90 for the Quito-318 instance; the local branching algorithm finds better solutions when k is set to larger than 90.

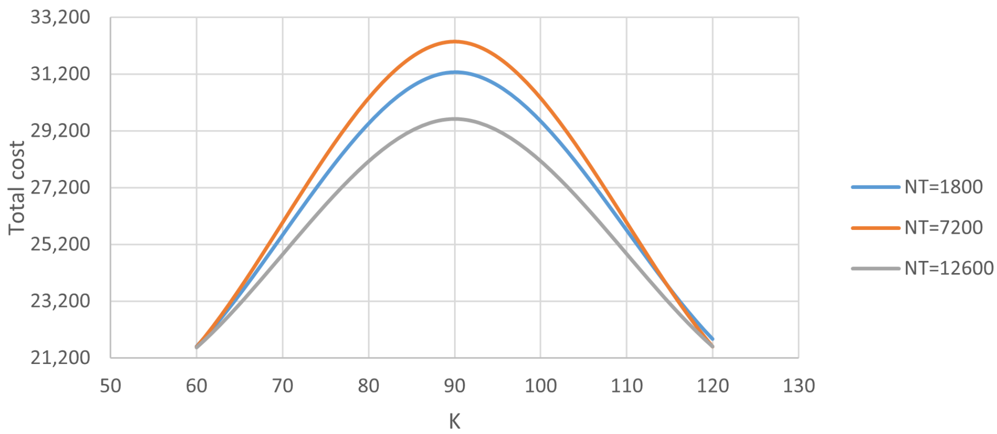


Figure 3.8 Comparison of the total cost of Quito-318 over various k and values for NT limit.

3.3 Concluding Remarks

In this chapter, we propose a local branching algorithm as an alternative to solving the problem directly with commercial solvers such as Gurobi. We provide extensive computational results to demonstrate the performance of the algorithm in comparison to Gurobi.

We evaluate the quality of the solutions obtained by the algorithm against Gurobi for the three real-life systems. Based on the size of the instances, our preliminary observations are as follows.

- The results on small instances imply that the algorithm finds the optimal solution in the same computational time as Gurobi.
- The results on medium instances indicate that the algorithm finds a near-optimal solution that competes with the solution obtained by Gurobi but in an extremely shorter computation time.

- Finally, we observe that obtaining the optimal solution in the large instance is still challenging, but good-quality solutions can be obtained within reasonable time.

Given that the local branching finds solutions almost as good as the solver, we conduct further experiments to study the solution evolution of the algorithm over time. Results show that the local branching finds a high-quality solution in early iterations, and it is mostly constant in the future iterations.

For the local branching algorithm, two parameters, i.e., k and the NT limit, may affect the quality of the solution or the computational time. In order to find an appropriate set of parameter values that lead to an efficient solution in terms of solution time or quality, we conduct extensive experiments with alternative values of both k and the NTl limit. For k, we observe a difference in the solution quality of medium instances; however, there is no noticeable difference in large instances with different k values. We also solve the MPLPP-VR for different NT limits. Similar to k, the NT limit may not significantly impact the local branching algorithm performance.

4. BENDERS AND LOGIC-BASED BENDERS DECOMPOSITION FOR SOLVING THE MPLPP-VR

In Chapter 3, we demonstrate that the local branching algorithm solves the small and medium instances to optimality in shorter computational times than Gurobi. In the largest instance, Quito-318, although the algorithm fails to find the optimal solution, the results indicate that it yields a high-quality solution, not necessarily better than but close to that of Gurobi, within shorter computation time. We further analyze the local branching parameters to improve the efficiency of the algorithm. Nevertheless, there are no significant gains from various alternative values of either k or the NT limit in terms of the computation times or the quality of the solutions.

There exist many attempts in the literature to mitigate the complexity of the optimization problems by iterative decomposition techniques. Dantzig–Wolfe decomposition (DWD) and BD are two prolific exact methods for solving NP-hard problems efficiently. DWD is applicable for optimization problems for which the complicating constraints disclose a decomposition into small subproblems (SPs) or pricing problems, along with a master problem (MP) which are considerably tractable (Vanderbeck & Savelsbergh, 2006). DWD usually requires the use of the Column Generation (CG) approach since the problem begins with extremely many variables (columns). The pricing problems are mainly convertible to standard optimization problems where one can apply efficient algorithms to solve them in a reasonable amount of time.

On the other hand, the BD (or variable partitioning) is appropriate for optimization problems with complicating variables of which, when they are fixed, the remaining problem is easy to solve (J.F.Benders, 1962). The BD decomposes the original problem into a restricted MP and a continuous linear SP. Based on the dual information from the solution(s) the SP(s), various valid inequalities are iteratively added to the MP to reach the optimal solution. In Section 4.1.1.1, we describe the classical BD algorithm in detail.

As mentioned in Chapter 2, we have three types of decision variables in the MPLPP-

VR: binary decision variables for line selection decisions, i.e., y_l, and integer decision variables for frequency of the lines, v_l^t, and the number of transfers between each pair of lines, $w_{lk}^{t_i t_j}$. We suppose that if line selection decisions are fixed, the remaining problem to determine the frequencies to the lines and the possible number of transfers between each selected set of lines can be solved in a substantially short computation time.

Therefore, the decomposition is applied by partitioning the constraints related to y into an MP and the remaining constraints along with the relevant decision variables to an SP. At each iteration, the MP is solved to optimality, i.e., defining the optimal set of lines, and accordingly, either a feasibility or an optimality cut is derived from the dual solution of the SP; otherwise, the latest set of lines is an optimal solution.

The remainder of this chapter is organized as follows. In Section 4.1, we provide a review of the classical BD. In Section 4.1.1, we present our implementation of the classical BD scheme and supplement it with computational results. A new LBBD and computational experiments are presented in Section 4.1.2. Finally, in Section 4.1.2.3, we develop our second LBBD and demonstrate the results. Finally, in Section 4.2, we provide the concluding remarks.

4.1 Benders Decomposition

BD is an exact algorithm successfully implemented in a broad range of optimization problems; it is originally introduced to solve linear MIP problems. The original problem is decomposed into two simpler problems: an MP involving the complicating variables, e.g., integer variables and their associated constraints and an SP with continuous variables and remaining constraints (J.F.Benders, 1962). BD is an iterative procedure that solves both the MP and the SP once in each iteration. At each iteration, the values of complicating variables are first found by solving the MP; then, the SP is solved by fixing the values of those variables. If the SP is infeasible, a feasibility cut is added to the MP to exclude the corresponding infeasible solution; otherwise, the SP is feasible and provides the current extreme point solution by generating an optimality cut for the MP. The algorithm terminates when the gap between the solutions of the MP and the SP are less than or equal to a predetermined value or the number of iterations reaches a predefined value. Many promising extensions have been developed to enhance the efficiency of the algorithm and accelerate its convergence since the original BD may not be computationally tractable. Although most acceleration approaches are successful in different optimization problems, there are still many hurdles to leap such as selecting solutions for both the MP and the SP, feeding an initial solution as a warm-start, and generating strong cuts. Rahmaniani, Crainic, Gendreau & Rei (2017) present a taxonomy for both algorithmic improvements and convergence acceleration.

The BD has been extensively employed for solving network-related problems. Cordeau, Pasin & Solomon (2006) apply a BD approach to logistics network design problems. Since the subproblem solution is degenerate, they enhance the proposed BD algorithm by generating the Pareto optimal cuts. Fortz & Poss (2009) develop a BD algorithm for bi-layer networks such as telecommunication networks. Since the MP is computationally difficult, they use a B& B framework to add feasibility cuts to the MP. Computational experiments show that adding cuts through the B&B framework improves the solution time of the MP significantly. Errico, Crainic, Malucelli & Nonato (2017) develop a Branch-and-Cut procedure based on the BD for semi-flexible transit systems. They demonstrate that the proposed approach significantly outperforms a commercial solver. Based on the satisfactory results, starting from the traditional BD, we develop two new LBBDs.

4.1.1 Classical BD

We present details of our proposed classical BD approach for the MPLPP-VR along with the proposed algorithm. Further, we develop different acceleration methods to improve the convergence rate.

4.1.1.1 Classical BD framework

To solve the MPLPP-VR, let us consider the decision variables y_l as the complicating variables. By fixing the decision variables y_l to given values $\overline{y}_l$ for each $l \in l$, the SP in the MPLPP-VR is formulated as

$$\text{(4.1)} \qquad \text{minimize} \quad \sum_{l \in L} \sum_{t \in T} c_l^s v_l^t$$

$$(2.14),(2.16)\text{-}(2.22).$$

The corresponding dual of the SP (DSP$_{MPLPP-VR}$) can be stated as

$$\text{(4.2)} \qquad \text{minimize} \sum_{e \in E_l} \sum_{t \in T} d_e^t \gamma_e^t - \sum_{l \in L} \sum_{t \in T} \mathcal{W} \alpha_l^t \overline{y}_l + (\tau + \sigma)\mathcal{U}$$

$$\text{(4.3)} \qquad \text{subject to} \sum_{e \in E_l} \mathcal{K} \gamma_e^t - \alpha_l^t - \beta_l^t \le c_l^s \qquad \forall l \in L, \forall t \in T,$$

$$\beta_l^t + \xi_l^t - \xi_k^{t-\rho_{kl}} \le 0 \qquad \forall l,k \in L, \forall t \in T,$$

$$\text{(4.4)} \qquad t + \rho_{kl} \le |T|,$$

$$\text{(4.5)} \qquad \beta_l^t + \xi_l^t + \tau \le \qquad \forall k \in L, \forall t \in T,$$

$$\text{(4.6)} \qquad \xi^t + \sigma \le 0 \qquad \forall k \in L, \forall t \in T,$$

$$\text{(4.7)} \qquad \gamma_e^t \ge 0 \qquad \forall e \in E, \forall t \in T,$$

$$\text{(4.8)} \qquad \alpha_l^t \ge 0 \qquad \forall l \in L, \forall t \in T,$$

where for all $e \in E$ and $t \in T$, γ_e^t denotes the dual variables related to the demand coverage constraints (2.14) where α_l^t, β_l^t, and ξ_l^t, are the dual variables corresponding to the constraints (2.15), (2.16), and (2.17) for all $l \in L$ and $t \in T$, respectively. Finally, τ and σ are dual variables that correspond to constraints (2.18), and (2.19), respectively. According to the solution of the DSP$_{MPLPP-VR}$ and generated valid cuts, we can formulate the MP$_{MPLPP-VR}$ as

$$\text{(4.9)} \qquad \text{minimize} \quad \sum_{l \in L} c_l^f y_l + \eta$$

$$\text{(4.10)} \qquad \text{subject to} \quad \sum_{e \in E_l} \sum_{t \in T} d_e^t \gamma_e^t - \sum_{l \in L} \sum_{t \in T} \mathcal{W} \alpha_l^t \overline{y}_l + \tau \mathcal{U} + \sigma \mathcal{U} \le 0,$$

$$\text{(4.11)} \qquad \sum_{e \in E_l} \sum_{t \in T} d_e^t \gamma_e^t - \sum_{l \in L} \sum_{t \in T} \mathcal{W} \alpha_l^t \overline{y}_l + \tau \mathcal{U} + \sigma \mathcal{U} \le \eta,$$

where η is the decision variable representing the lower estimator of the

DSP$_{MPLPP-VR}$. At each iteration, the DSP$_{MPLPP-VR}$ obtains an upper-bound and the MP$_{MPLPP-VR}$ provides a lower-bound for the original problem.

The main drawback of the classical BD is its slow convergence rate (Rahmaniani et al., 2017). Different acceleration approaches are suggested in the literature. Since our initial results also show the slow convergence rate of the algorithm, we apply the following acceleration approaches to our BD algorithm:

- Pareto-optimal cut,
- single search tree, and
- covering cut bundle generation.

When the SP is a network optimization problem, it is common to obtain degenerate solutions which means the DSP has multiple optimal solutions. In Chapter 2, we show that the vehicle transfer constraints can also be represented with a network flow formulation. These constraints correspond to the SP. In this respect, the SP can be interpreted as a network flow problem. Applying a strong (Pareto-optimal) cut among many possible valid optimality cuts may decrease the number of iterations and consequently improve the convergence rate. Magnanti & Wong (1981) show that a Pareto-optimal cut is not dominated by any other cuts. To define a Pareto-optimal cut, they suggest using the core point of the MP. The core point is a point in the relative interior of the convex hull of the feasible region of the MP. Thus, to generate a Pareto-optimal cut, we need to solve an auxiliary problem as

$$\text{(4.12)} \quad \text{maximize} \quad \sum_{e\in E_l}\sum_{t\in T} d_e^t\gamma_e^t - \sum_{l\in L}\sum_{t\in T}\mathcal{W}\alpha_l^t y_l^0 + \tau\mathcal{U} + \sigma\mathcal{U}$$

$$\text{(4.13)} \quad \text{subject to} \quad \sum_{e\in E_l}\sum_{t\in T} d_e^t\gamma_e^t - \sum_{l\in L}\sum_{t\in T}\mathcal{W}\alpha_l^t \overline{y}_l + \tau\mathcal{U} + \sigma\mathcal{U} = DSP_{MPLPP-VR}(\overline{y}),$$

$$\text{(4.3)- (4.8)},$$

where y^0 is the core point of the MP$_{MPLPP-VR}$.

Finding the core point is difficult and time-consuming at each iteration. Papadakos (2008) proves that y^0 can be any point in the feasible region of the MP. The feasible region of the auxiliary problem is the same as the original problem while the objective functions are different. Thus, using any point in the feasible region still generates a valid optimality cut although it is not necessarily a Pareto-optimal cut. Papadakos (2008) updates the core point without solving the auxiliary problem at iteration i by the equation

$$y_i^0 = 0.5y_{i-1}^0 + 0.5y_i^0. \tag{4.14}$$

The initial core point can be one of the following points (Maher, 2021):

(i) relative interior point,

(ii) the first primal feasible solution,

(iii) the first linear programming solution,

(iv) solution vector of ones, and

(v) solution vector of zeros.

At each iteration, for the $\text{MP}_{MPLPP-VR}$, a new branch-and-bound tree is built and solved to find the optimal solution. In this strategy, considerable time is spent revisiting candidate solutions that have been removed earlier. Alternatively, we can create a single search tree and generate valid cuts for the integer solutions found inside the search tree. To generate a single search tree, we can use the callback function as a modern re-optimization tool in Gurobi. In this method, we generate a cut when a new incumbent solution is found. Thus, it is a trade-off between generating a cut at each node of the search tree and the computational effort for solving a large $\text{MP}_{MPLPP-VR}$. To overcome this difficulty, we use the lazy constraints technology (of the solver Gurobi and keep all valid cuts in a cut pool. With this strategy, at each node, we recheck all cuts in the pool for the current incumbent solution and add those cuts which are violated.

In the covering cut bundle generation, at each iteration, a number of valid cuts, say a bundle of low-density cuts, are added to the $\text{MP}_{MPLPP-VR}$ instead of one low-density cut. A cut is low-density if a small number of decision variables have positive coefficients. At each iteration, we check the cut which is generated by the classical BD and determine which variables are not covered by the BD cut. Next, we develop a new low-density cut in which at least one of the uncovered variables in the previous cut is covered. The algorithm terminates if a predefined number of decision variables are covered; for more details, refer to Saharidis & Ierapetritou (2013).

4.1.1.2 Computational experiment for the classical BD

We apply both the classical BD and the BD with different acceleration approaches to solve the MPLPP-VR for all instances. The aforementioned acceleration approaches, i.e., a single search tree, covering cut bundle, and adding Pareto-optimal cuts, are considered.

Table 4.1 shows the results. Since the behavior of the BD for all three daily demand data (i.e., Weekday, Saturday, and Sunday) are similar in the Istanbul Metrobüs system, we present the results of Weekday for the sake of simplicity. The number of iterations is the total number of times that the BD solves the MP and the DSP, the solution time is the total time that the BD searches the solution space, and the total cost is the sum of the fixed cost and the operation cost.

The results show that using Pareto-optimal cuts as an acceleration approach improves the convergence of the BD. In the Istanbul Metrobüs system, although the solution times for all acceleration approaches are already short, adding the Pareto-optimal cuts to the MP decreases the number of iterations. For the Athens system, by using Pareto-optimal cuts, the number of iterations decreases by approximately 60% compared to the classical BD; it decreases by approximately 80% while we solve the MPLPP-VR with a single search tree. The performance of the BD with different acceleration approaches is highlighted through comparison with Gurobi. The results show that while the BD with Pareto-optimal cuts finds the optimal solution in the Istanbul Metrobüs system and the Athens, the computational times in both instances are significantly larger than the time spent by Gurobi.

Table 4.1 A Comparison among acceleration approaches of the BD ($\mathcal{K}$=250, $\mathcal{U}$=200, $\mathcal{W}$=36).

Solution method	Istanbul-Weekday			Athens		
	Number of iterations	Solution time	Total cost	Number of iterations	Solution time	Total cost
Gurobi	-	0.32	61334.60	-	1254	68030.58
BD-classical	9	1.09	61334.60	2128	86400	67679.14
BD-CCB	8	1.55	61334.60	218	26000	68312.99
BD-Pareto-optimal	8	1.55	61334.60	795	28768	68030.58
BD-Single search tree	18	1.88	61334.60	3892	86400	68099.92

Figures 4.1 and 4.2 represent the evolution of the gap between the objective function value of the SP and the MP while we solve the MPLPP-VR with alternative acceleration approaches for the Istanbul Metrobüs and the Athens instances. The results show that, for the Istanbul Metrobüs system, the classical BD finds an incumbent solution with a lower BD gap faster than other proposed acceleration approaches. However, for the Athens system, the BD finds the optimal solution only when using Pareto-optimal cuts.

For Quito-122 and Quito-318 instances, we do not observe a reasonable convergence within 48 hours time limit. As a result, we do not present the computational results for these instances.

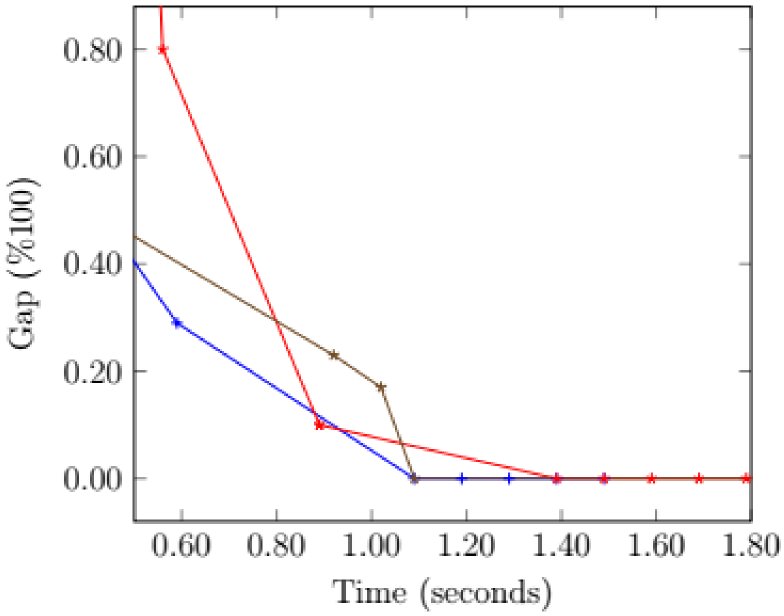


Figure 4.1 Evolution of the gap for the different acceleration approaches- Istanbul (K=250, U=200, W=36).

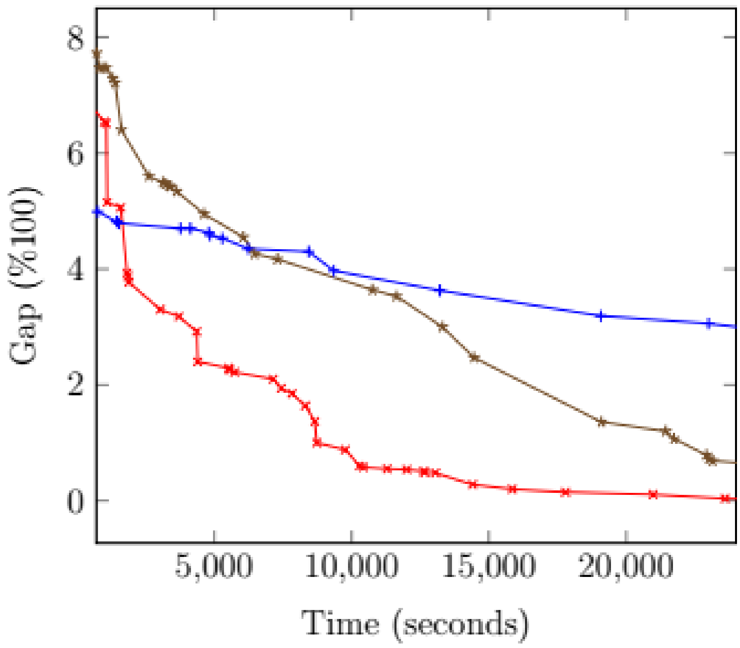


Figure 4.2 Evolution of the gap for the different acceleration approaches- Athens (K=250, U=200, W=36).

4.1.2 Logic-based Benders Decomposition

As mentioned earlier, in the classical BD, since we exploit the dual information of the SP, it needs to be in the form of continuous linear programming (J.F.Benders, 1962). Typically, the SP in a network optimization problem has totally unimodular (TU) features. Therefore, in the absence of continuous linear formulation, we are still able to utilize the dual of the SP. However, the computational experiments reveal that the classical BD does not yield promising results even with applying different acceleration approaches.

Many state-of-the-art approaches have been developed to obviate the challenges regarding the necessity of a linear SP. LBBD has recently been adapted to solving such problems for which the SP has not necessarily a continuous linear form (Hooker, 2007). It has been employed to a range of optimization problems, including transportation network design (Peterson & Trick, 2009) planning and scheduling problems (Harjunkoski & Grossmann, 2002; Hooker, 2007), vehicle routing problem (Raidl, Baumhauer & Hu, 2014), and multi-period network interdiction (Enayaty-Ahangar, Rainwater & Sharkey, 2019).

4.1.2.1 LBBD with constraint propagation framework

We propose an LBBD algorithm with constraint propagation (LBBD-CP) to solve the MPLPP-VR. Constraint propagation is an indispensable technique to solve the problem of satisfying a set of constraints (Rossi, Van Beek & Walsh, 2006). Our decomposition scheme is inspired by Pferschy & Staněk (2017). They relax subtour elimination constraints from the original model, and the remaining problem is solved by the ILP solver. They iteratively add the subtour elimination constraints to the model when they are needed.

In the MPLPP-VR, balance constraints (2.17) are considered as the complicating constraints. We observe that the remaining problem may require less computational effort when constraints (2.17) are relaxed. Therefore, the MP is a relaxation of the MPLPP-VR, which does not include any restrictions on the vehicle transfers and their balance in both time and space. We refer to the MP as the line planning subproblem (LPsP). On the other hand, the SP scrutinizes whether the MP solution is balanced with respect to the vehicle transfers. Therefore in the proposed LBBD, the SP is a feasibility checking problem. We refer to the SP as the vehicle transfer

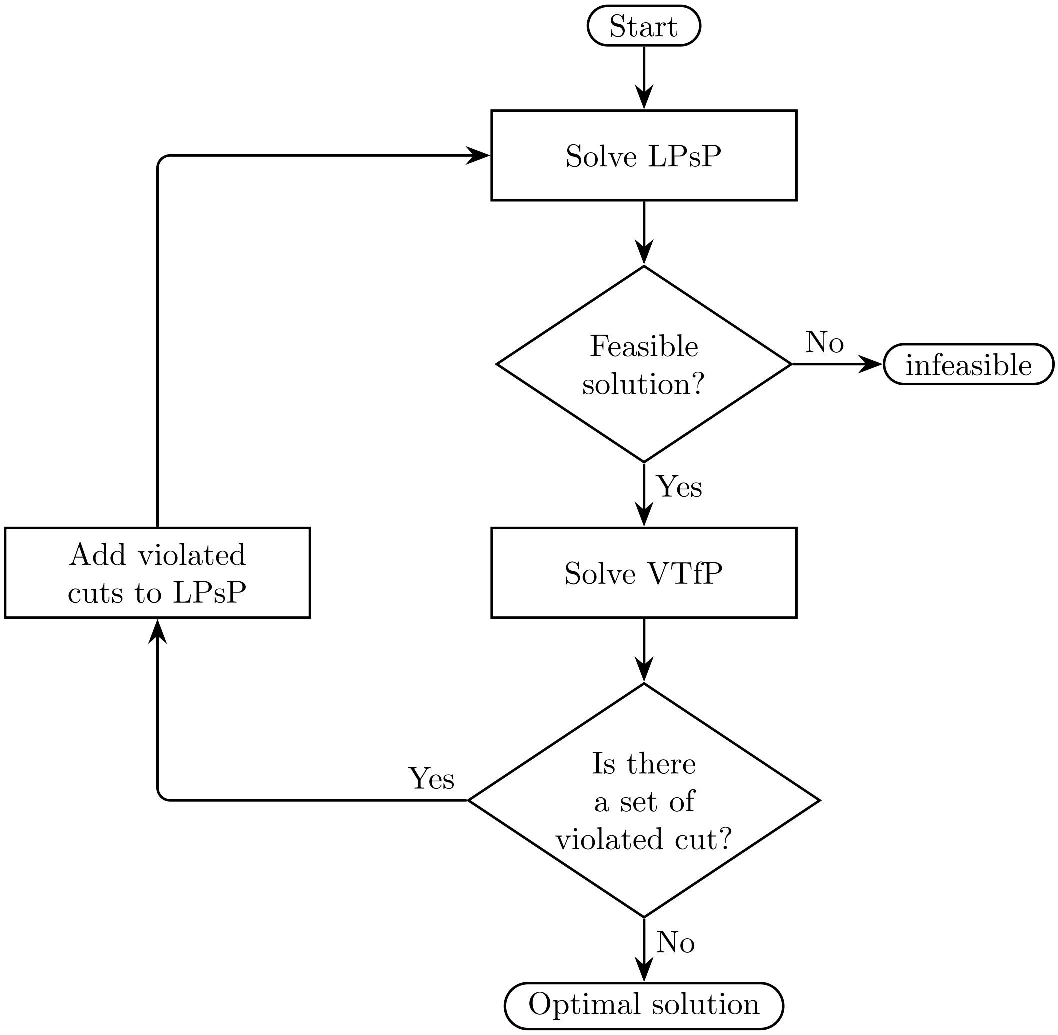


Figure 4.3 The flowchart for the LBBD-CP algorithm.

feasibility problem (VTfP).

Algorithm 2 An algorithm to generate the violated constraints for the VTfP

Require: MP solution, $Cons = \{\}$, $Violatedcuts = \{\}$;
Ensure: All Constraints (2.17) are satisfied;

$Cons \leftarrow$ Left hand sides (LHSs) of Constraints (2.17);
while $Cons$ **do**
  if $\sum_{\substack{k\in L_0 \\ t-\bar{\rho}_{kl}\geq 0}} w_{kl}^{t-\bar{\rho}_{kl},t} - \sum_{\substack{k\in L_0 \\ t+\bar{\rho}_{kl}\leq \boldsymbol{\mathcal{T}}}} w_{lk}^{t,t+\bar{\rho}_{lk}} \neq 0$ **then**

    $Violatedcuts \leftarrow \sum_{\substack{k\in L_0 \\ t-\bar{\rho}_{kl}\geq 0}} w_{kl}^{t-\bar{\rho}_{kl},t} - \sum_{\substack{k\in L_0 \\ t+\bar{\rho}_{kl}\leq \boldsymbol{\mathcal{T}}}} w_{lk}^{t,t+\bar{\rho}_{lk}} = 0$;
  else
    $Cons \leftarrow Cons - LHS$;
  end if
end while
return $Violatedcuts$

At each iteration, the LPsP is solved by a commercial solver. Then, the optimal solution of LPsP is checked for balance feasibility. To solve the VTfP, we use a feasibility checking algorithm. The iterative checking procedure is demonstrated in Algorithm 2. There are two possibilities: (i) VTfP is infeasible; accordingly, a set of violated constraints are added to the LPsP to exclude the current solution, or (ii) the VTfP is feasible. Figure 4.3 shows a schematic description of the LBBD-CP algorithm. The LBBD-CP terminates when no violated balance constraints exist. In this case, the LPsP solution is considered an optimal solution to the original problem.

4.1.2.2 Computational experiment for the LBBD with constraint propagation

To study the performance of the LBBD-CP in comparison to Gurobi, we first analyze the results for all instances provided in Section 2.4. From Table 4.2, the LBBD-CP solves all instances, except Quito-318 to optimality. In the Athens instance, we observe that the LBBD-CP requires only 417 seconds to obtain the optimal solution with an approximate 80% improvement in the CPU run time. In Quito-122, the LBBD-CP provides a huge improvement in solution time over Gurobi. The results from Table 4.2 show that LBBD-CP solves the problem in 9509 seconds while Gurobi needs 59135 seconds which is about an 84% improvement in the CPU run time. For Quito-318, however, our proposed algorithm is still effective and able to find a more competent feasible solution regarding the total cost within the same time limit. Since solving the LPsP to optimality requires too much time, we consider 3600 seconds as the time limit. Comparing the results of the local branching as shown in Table 3.1 with those of the corresponding LBBD-CP in Table 4.2, we observe that the LBBD-CP algorithm provides a significant improvement than the local branching algorithm compared to Gurobi.

Table 4.2 Results obtained by Gurobi and the LBBD-CP algorithm ($\mathcal{K}$=250, $\mathcal{U}$=200, $\mathcal{W}$=36).

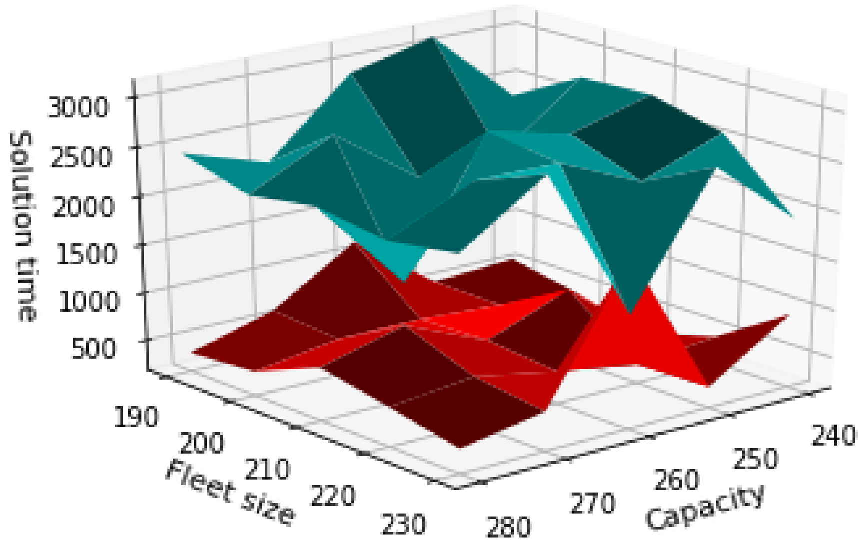
Instance	Gurobi		LBBD-CP	
	Cost	Time	Cost	Time
Istanbul-Weekday	61334.60	< 1	61334.60	< 1
Istanbul-Saturday	48807.00	< 1	48807.00	< 1
Istanbul-Sunday	35377.80	< 1	35377.80	< 1
Athens	68030.58	2071	68030.58	417
Quito-122	21690.17	59135	21690.17	9509
Quito-318	21611.21	86400	-	86400

To verify the superiority of LBBD-CP, we analyze the performance of the LBBD-CP against Gurobi under different levels of the capacity parameters of the system, the fleet size ($\mathcal{U}$) and the vehicle capacity ($\mathcal{K}$). In Figure 4.4, each point on the surface corresponds to a unique pair of vehicle capacity and fleet size and the computation time required to find the optimal solution. Because we cannot find the optimal solution for Quito-318 within a reasonable time, we only consider the Athens and Quito-212 instances. From Figure 4.4, one can observe that the red surface (i.e., LBBD-CP) is under the green surface (i.e., Gurobi), meaning that LBBD-CP consistently outperforms Gurobi in different levels of capacity in the system.

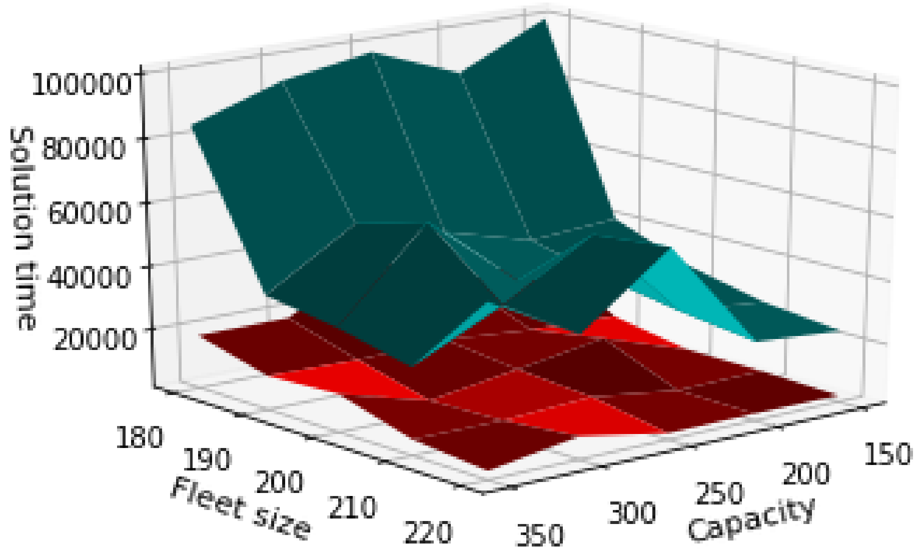
4.1.2.3 LBBD with max-flow cuts

The LBBD-CP decomposes the MPLPP-VR into two problems: an MPLPP-VR without the transfer balance constraints (2.17) and a feasibility SP for the transfer balance constraints. We illustrate that LBBD-CP substantially improves the computational time in small, medium and medium-to-large instances. Nevertheless, solving the largest instance, Quito-318, is still challenging, such that the LBBD-CP could not obtain a feasible solution within 86400 seconds (1 day).

An MPLPP-VR incorporates two optimization problems: an MPLPP that determines the lines and frequencies as addressed in Section 2.1 and a problem to deter-



(a) Athens



(b) Quito-122

Figure 4.4 Solution time with Gurobi (green surface) and LBBD-CP (red surface) for combinations of $\mathcal{K}$ and $\mathcal{U}$.

mine how vehicles are transferred between selected lines based on their frequencies, i.e. the solution of the MPLPP. In this respect, we propose an alternative decomposition scheme for another LBBD-CP implementation yielding an MP that selects a subset of lines with their frequencies by solving a generic MPLPP. The remaining SP provides a solution for the feasible transfer of vehicles based on the optimal solution of the MP. The corresponding SP is formulated as an optimization problem that minimizes the feasibility violations of constraints 2.16. For this purpose, σ_l^t and $\sigma_l^{'t}$ are defined as continuous variables denoting the excess or deficiency respectively in these constraints. Likewise, for each constraint 2.17, the continuous variables δ_l^t and $\delta_l^{'t}$ show the amount of constraint violations. The resulting SP, therefore, is formulated as

$$\text{(4.15)} \quad \text{minimize} \quad \sum_{l \in L} \sum_{t \in T} \sigma_l^t + \delta_l^t + \sigma_l^{'t} + \delta_l^{'t}$$

$$\text{(4.16)} \quad \text{subject to} \quad \sum_{\substack{k \in L_0 \\ t-\overline{\rho}_{kl} \geq 0}} w_{kl}^{t-\overline{\rho}_{kl},t} + \sigma_l^t - \sigma_l^{'t} = \overline{v_l^t} \qquad \forall l \in L, \forall t \in T,$$

$$\text{(4.17)} \quad \sum_{\substack{k \in L_0 \\ t-\overline{\rho}_{kl} \geq 0}} w_{kl}^{t-\overline{\rho}_{kl},t} - \sum_{\substack{k \in L_0 \\ t+\overline{\rho}_{kl} \leq \boldsymbol{\mathcal{T}}}} w_{lk}^{t,t+\overline{\rho}_{lk}} + \delta_l^t - \delta_l^{'t} = 0 \qquad \forall l \in L_0, \forall t \in T,$$

$$\text{(4.18)} \quad \sigma_l^t, \sigma_l^{'t} \geq 0 \qquad \forall l \in L, \forall t \in \cup T,$$

$$\text{(4.19)} \quad \delta_l^t, \delta_l^{'t} \geq 0 \qquad \forall l \in L_0, \forall t \in \cup T,$$

$$\text{(4.20)} \quad w_{lk}^{st} \in N \qquad \forall l \in L_0, \forall t \in \cup T,$$

$$\forall t \in T \cup \{\boldsymbol{\mathcal{T}} + 1\}, s < t.$$

where $\overline{v_l^t}$ denotes the frequencies from an MPLPP solution. Given the line frequencies from the optimal MP solution, denoted by $\overline{v_l^t}$, we solve the above problem to check if constraints associated with the transfer of vehicles are satisfied. If all constraints are satisfied, the optimal value of the objective function (4.15) of the SP is zero, and therefore the optimal solution of the MP is also optimal for the corresponding MPLPP-VR. Otherwise, the MP is extended with the set of selected violated constraints corresponding to non-zero slack or surplus variables. The algorithm iteratively continues in this fashion until the objective function value of the optimal solution for the SP is zero, i.e. no constraint violation is detected.

In the new LBBD-CP, however, the new SP is more challenging to solve while propagating constraints for the MP considerably increase the computational burden. Therefore, we develop an alternate constraint propagation procedure; the SP is redefined as a feasibility flow problem which can be solved as a maximum flow problem. We employ the max-flow min-cut theorem to derive the feasibility cuts to be fed to the MP, including information regarding the transfers. These cuts are iteratively incorporated into the MP. The algorithm terminates when we find a set of feasible transfers for the optimized solution of the MP.

For a given optimal solution of the MP, we construct a directed bipartite graph as follows:

- For each line $l \in L$ in each period $t \in T$, we create two nodes S_l^t (artificial supply node) and D_l^t (artificial demand node), where $t \in T \cup \{0, |T|+1\}$ and $l \in L_0$.
- For each period $t \in T$, we create a supply node and a demand node $S_{l_0}^t$ and a demand node $D_{l_0}^t$; we set their supply/demand amount as $\mathcal{U} - \sum_{l \in L} v_l^t$ if $l = l_0$.
- We create a supply node $S_{l_0}^0$ and set the supply amount to U; we create a demand node $D_{l_0}^0$ and set the demand amount to U.

All supply nodes are on the left-hand side of the bipartite graph while all demand nodes are on the right-hand side. The supply amount of the node S_l^t is $\overline{v_l^t}$ while the demand amount of the node D_l^t is also $\overline{v_l^t}$. If a vehicle can be transferred from line l to line k in ρ_{lk} periods, we create an uncapacitated arc from node S_l^t to node $D_k^{t+\rho_{lk}}$ as $(S_l^t, D_k^{t+\rho_{lk}})$. Figure 4.5 demonstrates the corresponding bipartite graph. If there exists a feasible flow in this bipartite graph, the optimal solution of the MP designated by $\overline{v_l^t}$ values is also optimal for the corresponding MPLPP-VR.

In order to find out if there exists a feasible flow in the bipartite graph, we solve a single-source-single-sink maximum flow problem on an expanded version of this bipartite graph as follows:

- We define a super-source node O and a super-sink node O'.
- From the super-source node O, we create an arc (O, S_l^t) to every supply node S_l^t, $l \in L$, with capacity equal to the original supply amount of the supply node, $\overline{v_l^t}$, and set the supply amount of the original supply node to 0.
- From each demand node D_l^t, $l \in L$, we create an arc (D_l^t, O') to the super-sink node O' with capacity equal to the original demand amount of the demand node, $\overline{v_l^t}$, and set the demand amount of the original demand node to 0.

Figure 4.6 shows the transformed bipartite graph.

Theorem 1 *A given directed bipartite graph has a feasible flow if and only the maximum flow in the transformed graph saturates all the source and the sink arcs (Ahuja, Magnanti & Orlin, 1993).*

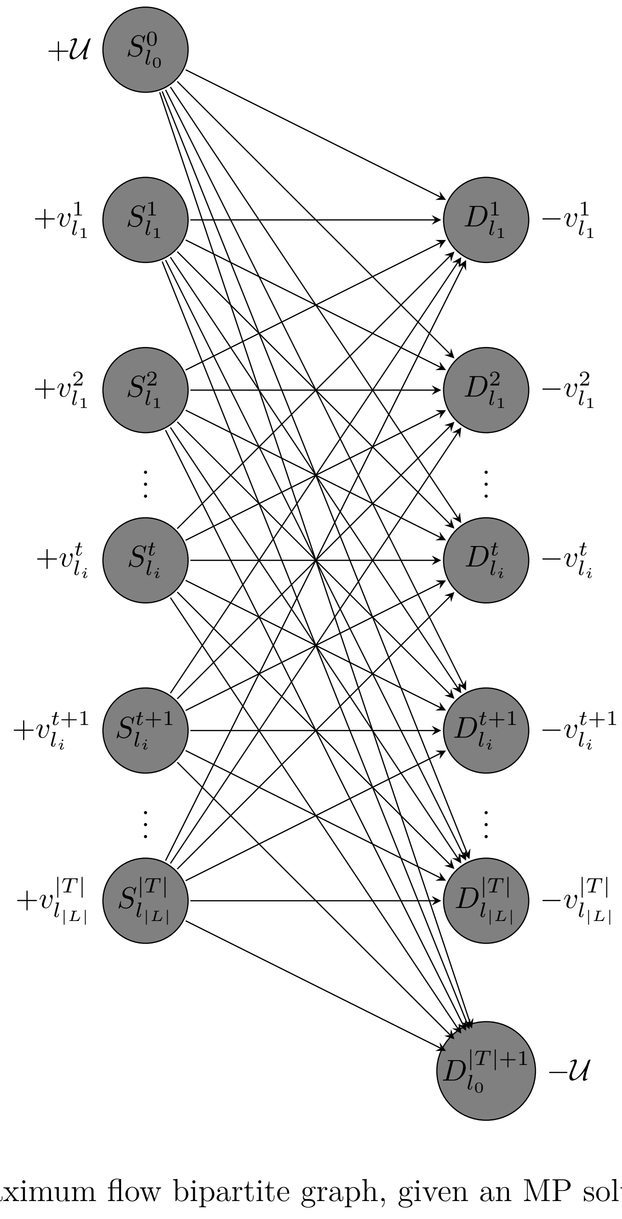


Figure 4.5 The maximum flow bipartite graph, given an MP solution

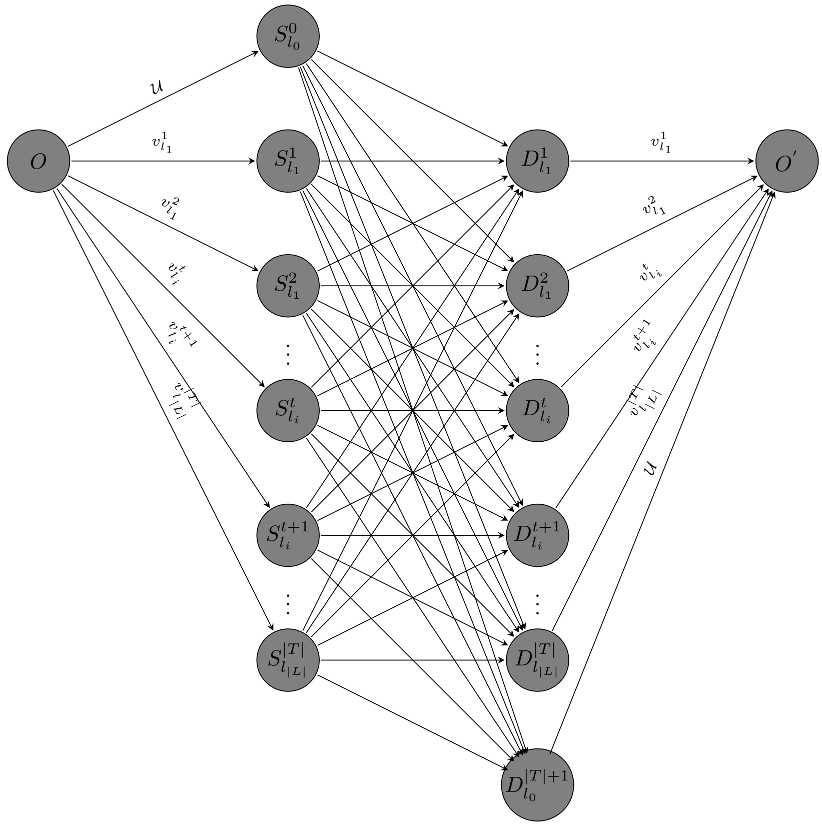


Figure 4.6 The maximum flow transformed bipartite graph, given an MP solution

At each iteration, the MP optimizes the line selection variables $\boldsymbol{y}^*$ and service level variables $\boldsymbol{v}^*$. The optimized solution of the MP is passed into the SP and result in a transformed bipartite graph, denoted by $G(\boldsymbol{y}^*, \boldsymbol{v}^*)$. Since our SP defines a standard maximum flow problem, we solve the SP via one of the well-known maximum flow algorithms. The SP determines whether the optimal solution of the MP leads to a feasible line plan with respect to the transfers.

We extend Theorem 1 in the context of our LBBD-max as the following theorem. In the proposed LBBD with maximum flow (LBBD-Max), the Benders cuts are derived based on the following theorem.

Theorem 2 *For a given optimal solution of the MP, $(\boldsymbol{y}^*, \boldsymbol{v}^*)$, if the maximum flow in transformed graph $G(\boldsymbol{y}^*, \boldsymbol{v}^*)$ saturates all arcs emanating from the source node or all arcs entering the sink node, $(\boldsymbol{y}^*, \boldsymbol{v}^*)$ satisfy all vehicle transfer constraints in the MPLPP-VR.*

The proof is straightforward. Since for each feasible transfer, an arc with positive capacity are defined, the solution of the SP satisfies all constraints associated with transfer variables, i.e., constraints (2.16)-(2.19). In addition, all arcs emanating from O are saturated, and we do not have any path to augment flow; therefore, $(\boldsymbol{y}^*, \boldsymbol{v}^*)$

determine a feasible set of transfers for MPLPP-VR.

Based on Theorem 2, if we observe a non-saturated arc emanating from the source node O or a non-saturated arc entering the sink node O', the solution of the MP is infeasible for the MPLPP-VR. Therefore, there exists a cut corresponding to $(\boldsymbol{y}^*, \boldsymbol{v}^*)$ which removes the current infeasible solution from the MP. When the maximum flow is strictly less than the total capacity of arcs emanating from the source node, Λ, we generate a cut as

$$\sum_{\substack{S^t_{l_j} \in \bar{S} \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{sS^t_{l_j}} \geq \sum_{\substack{D^t_{l_j} \in \bar{S} \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{D^t_{l_j} t} \tag{4.21}$$

based on the infeasibility of the flow on the bipartite graph. While developing (4.21), we benefit from the max-flow min-cut theorem. The theorem demonstrates that the maximum flow from the source node O to the sink node O' equals the capacity of the minimum cut (Ahuja et al., 1993). A cut $[\mathcal{S}, \bar{\mathcal{S}}]$ partitions a set of nodes into two subsets such that $O \in \mathcal{S}$ and $O' \in \bar{\mathcal{S}}$. An arc $(i, j) \in [\mathcal{S}, \bar{\mathcal{S}}]$ if $i \in S$ and $j \in \bar{S}$, or vice versa. The capacity of the cut $[\mathcal{S}, \bar{\mathcal{S}}]$ is

$$\text{Cap}([\mathcal{S}, \bar{\mathcal{S}}]) = \sum_{(i,j) \in [S, \bar{S}]} u_{ij} \tag{4.22}$$

where u_{ij} is a capacity associated with arc (i, j). Therefore, a minimum cut is a minimum capacity cut among all $[\mathcal{S}, \bar{\mathcal{S}}]$ cuts.

Theorem 3 *Inequality* (4.21) *removes the current MP solution* $\boldsymbol{v}^*$) *when added to the MP.*

It is sufficient to show that the capacity of the minimum cut should be strictly greater than the total capacity on source arcs, i.e., Λ. Knowing that in each cut $[\mathcal{S}, \bar{\mathcal{S}}]$

$$\sum_{\substack{D^t_{l_j} \in \bar{S} \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{D^t_{l_j} t} + \sum_{\substack{D^t_{l_j} \in S \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{D^t_{l_j} t} = \Lambda, \tag{4.23}$$

we can rewrite inequality (4.21) as

$$\sum_{\substack{S^t_{l_j} \in \bar{S} \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{sS^t_{l_j}} + \sum_{\substack{D^t_{l_j} \in \bar{S} \\ l_j \in L_0, t \in T \cup \{0, \boldsymbol{\mathcal{T}}+1\}}} u_{D^t_{l_j} t} \geq \Lambda. \tag{4.24}$$

The left-hand side of inequality (4.24) shows the capacity of the cut over the ingoing arcs to the sink node, and since,

$$\sum_{(i,j) \in [S, \bar{S}]} u_{ij} \geq \Lambda, \tag{4.25}$$

the cut removes the current MP solution. Algorithm 3 demonstrates a schematic description of the LBBD-max algorithm.

Algorithm 3 LBBD WITH MAXIMUM FLOW

Require: MP, SP, $MaxValue = 0$, $\Lambda = +\infty$
1: Call MIP solver to solve MP
2: Construct a bipartite graph $G(\boldsymbol{y}^*, \boldsymbol{v}^*)$ and Solve a maximum flow problem
3: Update $MaxValue$ and Λ
4: **if** $MaxValue < \Lambda$ **then**
5:     Add feasibility cut (4.21) to MP
6: **else**
7:     Break
8: **end if**
9: **return** $\boldsymbol{y}$, $\boldsymbol{v}$, and $\boldsymbol{w}$

4.1.2.4 Computational experiments for LBBD with max-flow cut

We study the performance of the LBBD-Max flow by comparing it against the LBBD-CP. In Table 4.3, the Cost column shows the best feasible solution found while the Time columns show the CPU time in seconds. The results with Metrobüs instances do not distinguish between the alternative decomposition approaches since the corresponding problems are already small enough to be solved to optimality in less than 1 second. Looking at the results for the Athens and Quito systems, we observe that the LBBD-Max algorithm in comparison to LBBD-CP improves the performance of the commercial solver significantly.

Table 4.3 Results obtained by the LBBD-CP and LBBD-Max algorithm ($\mathcal{K}$=250, $\mathcal{U}$=200, $\mathcal{W}$=36).

Instance	LBBD-CP		LBBD-Max	
	Cost	Time	Cost	Time
Istanbul- Weekday	61334.60	< 1	61334.60	< 1
Istanbul- Saturday	48807.00	< 1	48807.00	< 1
Istanbul- Sunday	35377.80	< 1	35377.80	< 1
Athens	68030.58	417	68030.58	3.45
Quito-122	21690.17	9509	21690.17	81.41
Quito-318	-	86400	21509.35	3605.83

In order to observe the performance of LBBD-Max when system resources are tighter, we change the vehicle capacity ($\mathcal{K}$), fleet size ($\mathcal{U}$), and the maximum number of vehicles ($\mathcal{W}$) to be assigned to a line. We avoid further investigation with Metrobüs instances since it takes less than 1 second to solve the related problems to optimality. Table 4.4 shows the corresponding results. The results for the Athens system show that LBBD-CP has inferior performance in comparison with Gruobi when the system capacity is tight as the solution time increases by about five times. In contrast, we observe approximately a 50% reduction in computational time in comparison with Gruobi when we solve the Athens system with the LBBD-Max algorithm. Since the optimal SP solution cannot be found for Quito-318 instance even at the first iteration, a time limit of 3600 seconds is set to solve the SP in each iteration, and the best feasible solution found within this limit is used to find feasibility cuts. The result in Table 4.4 shows that the LBBD-Max obtains a better feasible solution in much less computational time, approximately 5800 seconds, in comparison with Gurobi. As in Quito-122, the LBBD-CP cannot provide a feasible solution for Quito-318 within the same time limit as Gurobi.

Table 4.4 Results obtained by the LBBD-CP and LBBD-Max algorithm.

Instance	Parameters	Gurobi		LBBD-CP		LBBD-Max	
	($\mathcal{K}$,$\mathcal{U}$,$\mathcal{W}$)	Cost	Time	Cost	Time	Cost	Time
Athens	(235, 190, 20)	72656.30	221.30	72656.30	1065.49	72656.30	112.93
Quito-122	(150, 120, 20)	31432.90	72226	-	86400	31432.90	718.58
Quito-318	(150, 120, 15)	31301.02	86400	-	86400	30865.90	5404.65

The results in both Quito instances indicate a promising performance of the LBBD-Max. We observe that, in the Quito-122 instance, when the capacity is limited, the LBBD-CP fails to find even a feasible solution in 86400 seconds (1 day) while the LBBD-Max only requires 719 seconds to acquire the optimal solution. Further, when we compare the computational time of the LBBD-Max with Gruobi, it can be seen that the LBBD-Max algorithm improves the computational time by about

100%.

4.2 Concluding Remarks

In this chapter, to solve the MPLPP-VR, we first develop a classical BD algorithm. Since we observe the low convergence rate, we investigate different acceleration approaches such as the Pareto-optimal cut, a single search tree, and covering cut bundle generation. Results show that the classical BD has a low-grade convergence rate even by considering acceleration approaches.

While the results of the BD are not satisfactory in terms of the solution time, we develop an LBBD with constraint propagation. We compare Gurobi and LBBD-CP for different alternatives for system capacities. Results show that the new solution approach remarkably improves the solution time.

We show that while the system's capacity is tight, the LBBD-CP fails to find the optimal solution. In this respect, we design a second LBBD, in which the MP is an MPLPP and the SP is a feasibility problem regarding the transfer of vehicles between lines. To derive the feasibility cuts in the SP, we reformulate the vehicle transfer feasibility problem to a max-flow problem and resort to the max-flow min-cut theorem. Results show that the LBBD-Max finds the optimal solution fast enough when the capacity of the system is large enough. If the capacity of the system is tight, the LBBD-Max obtains the optimal solution quickly. In large instances, however, since solving the MPLPP is computationally challenging, we set a time limit, and therefore LBBD-Max finds a satisfactory feasible solution within reasonable computational time.

5. CONCLUSION AND FUTURE RESEARCH

The main focus of this work is a novel multi-period approach for the well-known line planning problem in the context of public transportation planning. As demand is the main input for the line planning problem, recognizing the its time-dependent behavior, we illustrate that urban transportation systems and associated plans are highly affected by fluctuations in transit demand during the day. To provide a conducive response to the demand-intensive environment, both temporal and spatial changes in demand should be considered at the line planning stage. With an effort to improve the demand-responsiveness of line plans, our primary contribution is developing a multi-period line planning model that considers the changes in transit demand over time.

In our cost-oriented multi-period approach, we consider fixed costs of line selection and variable operating costs depending on the service frequency on lines; the planning horizon is divided into discrete periods, each associated with a different demand pattern. We show that a multi-period approach outperforms a traditional single-period approach that combines line planning solutions of independent individual periods.

In order to obtain realistic and practical solutions, the assignment of resources to lines throughout the planning horizon is of paramount importance in the case of multi-period planning. We show that resource planning is mostly coupled with time periods. As the line frequencies change from one period to the next, the vehicles are to be reallocated or reassigned among the lines. In this respect, we develop a generalization of the first MPLPP formulation by integrating resource allocation and transfer constraints and exemplify with the MPLPP with vehicle rotation, MPLPP-VR. Our computational experiments with the MPLPP-VR show that solutions may change considerably when resources are tight in the system. Eventually, we also observe that choosing the period length may be a convoluted decision which is a trade-off between the accuracy of the solutions and the efficiency in resource planning.

We notice that computational challenges persist 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 the 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. Therefore, we consider alternative solution methods that can be scaled to solve multi-period line planning problems with vehicle transfers even for a very-large PTN.

To solve our problem efficiently, we propose a local branching algorithm. The local branching algorithm explores the feasible region by separating it into sub-regions with adding a local branching cut iteratively to the problem. We define the local branching cuts based on the line selection decision variables. Further, we develop extensive computational experiments to illustrate the algorithm's performance by comparing its results with the best solutions provided by Gurobi.

Our computational results in Chapter 3 indicate that the local branching algorithm beats Gurobi in terms of computational time, in small and medium size instances. On the other hand, the algorithm finds a high-quality solution in larger instances. The convergence behavior within the total time limit represents that the local branching algorithm finds a near-optimal solution in early iterations and gradually stabilizes in late iterations. We furthermore scrutinize the effect of the local branching parameters. Results verify that it is crucial to tune the crucial parameters beforehand in a way that the algorithm finds the optimal or near-optimal solutions in a reasonable run time.

We note that the local branching heuristic provides quite satisfactory performance as it finds the optimal solution for small instances and almost optimal solutions for medium instances within much shorter CPU time required to find the optimal solutions. Moreover, when the proposed algorithm fails to find the optimal solution, it usually enhances the computational efficiency by obtaining a solution with a smaller gap than the best feasible solutions identified by Gurobi. Nonetheless, we still need to develop an algorithm with better optimization performance, particularly in solving large instances. Therefore, we resort to the classical Benders decomposition and two implemetations of the Logic-based Benders decomposition.

First, we develop a classical implementation of BD to solve the MPLPP-VR. To generate promising feasibility/optimality cuts, we benefit from the total unimodularity characteristic of the sub-problem. However, due to the slow convergence rate, we adopt various acceleration approaches, including the strong (Pareto-optimal) cut, covering cut bundle generation, and single search tree. The performance of the proposed algorithm is evaluated by comparing it with Gurobi. The results show that using the Pareto-optimal cuts improves the convergence rate remarkably.

Second, we present two promising decomposition schemes to seek the optimal line plan with reasonable computational effort. The spirit of our first logic-based decomposition with constraint propagation approach dates back to the original ideas 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. In our decomposition approach, the first sub-problem is a multi-period line planning that does not include any restrictions on the vehicle transfer balance. In contrast, the second sub-problem investigates whether the current solution is balanced with respect to vehicle transfers. At each iteration, the optimal solution of the first sub-problem is checked for balance feasibility in the second sub-problem and the set of violated constraints is added (if it exists) to the first sub-problem. The algorithm continues until no violated constraints exist. Results show that good-quality solutions are obtained within a reasonable computational time.

Our computational experiments show that the logic-based decomposition with constraint propagation finds even better solutions than the local branching algorithm within the same CPU time limit. It seems plausible to employ the latest method for large-scale instances. We conduct further experiments on instances of the same problem set with different demand patterns and vary the problem parameters such as the capacity of the vehicles, the size of the fleet. Results verify that the decomposition algorithm outperforms Gurobi in different levels of capacity in the system.

We also develop an alternative logic-based decomposition scheme with which can benefit from the network optimisation algorithms, particularly the maximum flow algorithm. We decompose our problem into two sub-problems. In the first sub-problem, we solve a MPLPP without considering permissible transfers. Next, according to the MPLPP solution, a bipartite network is generated such that the capacities of the arcs are the line frequencies. The second sub-problem to be solved in each iteration is a standard maximum flow problem. The algorithm proceeds by adding a feasibility cut based on the max-flow min-cut to the first sub-problem iteratively. The algorithm terminates when the solution is feasible with respect to the permissible transfers. Results show a remarkable improvement in computational time.

This first effort to bring a multi-period planning approach into LPPs within the context of public transportation planning sheds light onto various modeling issues and computational aspects. It requires further analysis and understanding of the problem characteristics and the particular system, which opens up the venue primarily for sophisticated solution methods to solve the large instances of MPLPPs.

Challenged with expensive computational effort in larger instances, we may consider using the local branching algorithm to the solve the MP, which corresponds to an MPLPP. Earlier results show that the local branching algorithm performs well in solving the MPLPP-VR, it expected to show even a better performance for the MPLPP. We note that the algorithm finds the optimal or near-optimal solution within reasonable time if proper values are set for the algorithm parameters. In the LBBD-CP decomposition, the most time-consuming operation is identifying an optimal solution for the SP. Therefore, a hybridization of our proposed decomposition approach with the local branching algorithm may be a fruitful research idea to address the existing drawbacks in both algorithms.

Another stream of further research may focus on an extension of our mixed-integer linear programming problem formulation which explicitly considers the trip times and their effect on timely satisfaction of the transit demand. In the MPLPP-VR, we assume that a vehicle visits all edges corresponding to its trip path in the same period it is dispatched. If the effect of trip times is considered explicitly, a vehicle may not visit all edges in one period. Therefore, the effect of trip times on resource allocation is non-negligible, and characterizing its properties in the MPLPP-VR is also extremely essential to represent the relations between periods. This explicit consideration has two main challenges: changing the demand coverage constraints to satisfy the passenger demand accurately and modifying the flow conservation constraints to handle the vehicle transfers.

BIBLIOGRAPHY

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice hall.

An, K. & Lo, H. K. (2016). Two-phase stochastic program for transit network design under demand uncertainty. *Transportation Research Part B: Methodological*, *84*, 157–181.

Basso, L. J., Feres, F., & Silva, H. E. (2019). The efficiency of bus rapid transit (brt) systems: A dynamic congestion approach. *Transportation Research Part B: Methodological*, *127*, 47 – 71.

Ben-Ayed, O., Boyce, David, E., & Blair III, C. E. (1988). A general bilevel linear programming formulation of the network design problem. *Transportation Research Part B: Methodological*, *22*(4), 311–318.

Borndörfer, R., Arslan, O., Elijazyfer, Z., Güler, H., Renken, M., Şahin, G., & Schlechte, T. (2018). Line planning on path networks with application to the Istanbul Metrobüs. In *Operations Research Proceedings 2016*, (pp. 235 – 241).

Borndörfer, R., Grötschel, M., & Pfetsch, M. E. (2007). A column-generation approach to line planning in public transport. *Transportation Science*, *41*(1), 123–132.

Borndörfer, R., Reuther, M., Schlechte, T., & Weider, S. (2011). A Hypergraph Model for Railway Vehicle Rotation Planning. In *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, (pp. 146–155)., Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Bull, S. H., Rezanova, N. J., Lusby, R. M., & Larsen, J. (2016). *An applied optimization based method for line planning to minimize travel time*. DTU Management Engineering.

Bunte, S. & Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, *1*(4), 299–317.

Bussieck, M. (1998). *Optimal lines in public rail transport*. PhD thesis, Citeseer.

Bussieck, M. R., Kreuzer, P., & Zimmermann, U. T. (1997). Optimal lines for railway systems. *European Journal of Operational Research*, *96*(1), 54–63.

Cacchiani, V., Caprara, A., & Fischetti, M. (2012). A lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science*, *46*(1), 124–133.

Cadarso, L. & Marín, Á. (2012). Integration of timetable planning and rolling stock in rapid transit networks. *Annals of Operations Research*, *199*, 113–135.

Caprara, A., Monaci, M., & Toth, P. (2001). A global method for crew planning in railway applications. In *Computer-Aided Scheduling of Public Transport* (pp. 17–36). Springer.

Ceder, A. A. (2011). Public-transport vehicle scheduling with multi vehicle type. *Transportation Research Part C: Emerging Technologies*, *19*(3), 485–497.

Claessens, M., van Dijk, N. M., & Zwaneveld, P. J. (1998). Cost optimal allocation of rail passenger lines. *European Journal of Operational Research*, *110*(3), 474–489.

Cordeau, J.-F., Pasin, F., & Solomon, M. M. (2006). An integrated model for

logistics network design. *Annals of operations research*, *144*(1), 59–82.
D'ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European journal of operational research*, *183*(2), 643–657.
Enayaty-Ahangar, F., Rainwater, C. E., & Sharkey, T. C. (2019). A logic-based decomposition approach for multi-period network interdiction models. *Omega*, *87*, 71–85.
Errico, F., Crainic, T. G., Malucelli, F., & Nonato, M. (2013). A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. *Transportation Research Part C: Emerging Technologies*, *36*, 324–338.
Errico, F., Crainic, T. G., Malucelli, F., & Nonato, M. (2017). A benders decomposition approach for the symmetric tsp with generalized latency arising in the design of semiflexible transit systems. *Transportation Science*, *51*(2), 706–722.
Farahani, R. Z., Miandoabchi, E., Szeto, W. Y., & Rashidi, H. (2013). A review of urban transportation network design problems. *European Journal of Operational Research*, *229*(2), 281–302.
Fischetti, M. & Lodi, A. (2003). Local branching. *Mathematical programming*, *98*(1), 23–47.
Fischetti, M., Polo, C., & Scantamburlo, M. (2004). A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. *Networks: An International Journal*, *44*(2), 61–72.
Fortz, B. & Poss, M. (2009). An improved benders decomposition applied to a multi-layer network design problem. *Operations research letters*, *37*(5), 359–364.
Freling, R., Huisman, D., & Wagelmans, A. P. (2003). Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, *6*(1), 63–85.
Freling, R., Wagelmans, A. P., & Paixão, J. M. P. (1999). An overview of models and techniques for integrating vehicle and crew scheduling. *Computer-aided transit scheduling*, *471*, 441–460.
Goerigk, M., Schachtebeck, M., & Schöbel, A. (2013). Evaluating line concepts using travel times and robustness. *Public Transport*, *5*(3), 267–284.
Goerigk, M. & Schöbel, A. (2013). Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers & Operations Research*, *40*(5), 1363–1370.
Goossens, J.-W. (2004). *Models and algorithms for railway line planning problems.* PhD thesis.
Goossens, J.-W., van Hoesel, S., & Kroon, L. (2004). A branch-and-cut approach for solving railway line-planning problems. *Transportation Science*, *38*(3), 379–393.
Guihaire, V. & Hao, J.-K. (2008). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, *42*(10), 1251–1273.
Guo, X., Sun, H., Wu, J., Jin, J., Zhou, J., & Gao, Z. (2017). Multiperiod-based timetable optimization for metro transit networks. *Transportation Research Part B: Methodological*, *96*, 46–67.
Haase, K., Desaulniers, G., & Desrosiers, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation science*, *35*(3), 286–303.

Hanssmann, F. & Hess, S. W. (1960). A linear programming approach to production and employment scheduling. *Management Technology*, *1*(1), 46–51.

Harbering, J. (2013). *Planning a Public Transportation System with a View Towards Passengers*. PhD thesis, Georg-August-Universität Göttingen.

Harbering, J., Schiewe, A., & Schöbel, A. (1999). Lintim - integrated optimization in public transportation.

Harjunkoski, I. & Grossmann, I. E. (2002). Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. *Computers & Chemical Engineering*, *26*(11), 1533–1552.

Heil, J., Hoffmann, K., & Buscher, U. (2020). Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research*, *283*(2), 405–425.

Holt, C. C., Modigliani, F., & Simon, H. A. (1955). A linear decision rule for production and employment scheduling. *Management Science*, *2*(1), 1–30.

Hooker, J. N. (2007). Planning and scheduling by logic-based benders decomposition. *Operations research*, *55*(3), 588–602.

Horváth, B. (2012). Forecasting of travel demand in urban public transport. In *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, (pp. 317–321). IEEE.

Hsu, J. (1977). Transit routing problem(preliminary draft).

Ibarra-Rojas, O. J., López-Irarragorri, F., & Rios-Solis, Y. A. (2016). Multiperiod bus timetabling. *Transportation Science*, *50*(3), 805–822.

Ibarra-Rojas, O. J. & Rios-Solis, Y. A. (2011). Integrating synchronization bus timetabling and single-depot single-type vehicle scheduling. In *ORP3 Meeting, Cadiz*.

J.F.Benders (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*, 238–252.

Jiang, X., Zhang, L., & Chen, X. M. (2014). Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in china. *Transportation Research Part C: Emerging Technologies*, *44*, 110–127.

Karbstein, M. (2016). Integrated line planning and passenger routing: Connectivity and transfers. In *Operations Research Proceedings 2014*, (pp. 263 – 269).

Kaspi, M. & Raviv, T. (2013). Service-oriented line planning and timetabling for passenger trains. *Transportation Science*, *47*(3), 295–311.

Kepaptsoglou, K. & Karlaftis, M. (2009). Transit route network design problem. *Journal of transportation engineering*, *135*(8), 491–505.

Kim, J.-U. & Kim, Y.-D. (1999). A decomposition approach to a multi-period vehicle scheduling problem. *Omega*, *27*(4), 421–430.

Klier, M. J. & Haase, K. (2008). Line optimization in public transport systems. In Kalcsics, J. & Nickel, S. (Eds.), *Operations Research Proceedings 2007*, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kroon, L., Maróti, G., Helmrich, M. R., Vromans, M., & Dekker, R. (2008). Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, *42*(6), 553–570.

Kroon, L. G., Peeters, L. W. P., Wagenaar, J. C., & Zuidwijk, R. A. (2014). Flexible connections in pesp models for cyclic passenger railway timetabling. *Transportation Science*, *48*(1), 136–154.

Legato, P. & Trunfio, R. (2014). A local branching-based algorithm for the quay crane scheduling problem under unidirectional schedules. *4OR*, *12*(2), 123–156.

Li, S., Xu, R., & Han, K. (2019). Demand-oriented train services optimization for a congested urban rail line: integrating short turning and heterogeneous headways. *Transportmetrica A: Transport Science*, *15*(2), 1459–1486.

Liebchen, C. & Möhring, R. H. (2007). The modeling power of the periodic event scheduling problem: railway timetables—and beyond. In *Algorithmic methods for railway optimization* (pp. 3–40). Springer.

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., & Prigge, A. (2010). Computing delay resistant railway timetables. *Computers & Operations Research*, *37*(5), 857–868.

Lo, H. K., An, K., & Lin, W.-h. (2013). Ferry service network design under demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, *59*, 48–70.

Lusby, R. M., Larsen, J., & Bull, S. (2018). A survey on robustness in railway planning. *European Journal of Operational Research*, *266*(1), 1–15.

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

Maher, S. J. (2021). Implementing the branch-and-cut approach for a general purpose benders' decomposition framework. *European Journal of Operational Research*, *290*(2), 479–498.

Malucelli, F., Nonato, M., & Pallottino, S. (1999). Demand adaptive systems: some proposals on flexible transit. In *Operational research in industry* (pp. 157–182). Springer.

Manitz, J., Harbering, J., Schmidt, M., Kneib, T., & Schöbel, A. (2017). Source estimation for propagation processes on complex networks with an application to delays in public transportation systems. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *66*(3), 521–536.

Maróti, G. (2006). Operations research models for railway rolling stock planning.

Niu, H., Zhou, X., & Gao, R. (2015). Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints. *Transportation Research Part B: Methodological*, *76*, 117–135.

Odijk, M. A. (1994). *Construction of Periodic Timetables. Pt. 1. A Cutting Plane Algorithm*. TU Delft.

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, *36*(4), 444–449.

Pätzold, J., Schiewe, A., Schiewe, P., & Schöbel, A. (2017). Look-ahead approaches for integrated planning in public transportation. In *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Pätzold, J. & Schöbel, A. (2016). A matching approach for periodic timetabling. In *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Petersen, H. L., Larsen, A., Madsen, O. B., Petersen, B., & Ropke, S. (2013). The simultaneous vehicle scheduling and passenger service problem. *Transportation Science*, *47*(4), 603–616.

Peterson, B. & Trick, M. A. (2009). A benders' approach to a transportation network design problem. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, (pp. 326–327). Springer.

Pferschy, U. & Staněk, R. (2017). Generating subtour elimination constraints for the tsp from pure integer solutions. *Central European journal of operations research*, *25*(1), 231–260.

Pu, S. & Zhan, S. (2021). Two-stage robust railway line-planning approach with passenger demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, *152*, 102372.

Puhl, C. & stillerl, s. The maximum capacity of a line plan is inapproximable. *Technical report*, *60*.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817.

Raidl, G. R., Baumhauer, T., & Hu, B. (2014). Speeding up logic-based benders' decomposition by a metaheuristic for a bi-level capacitated vehicle routing problem. In *International Workshop on Hybrid Metaheuristics*, (pp. 183–197). Springer.

Rea, J. C. (1971). Designing urban transit systems: an approach to the route-technology selection problem. Technical report, University of Washington Seattle.

Rittner, M. & Nachtigall, K. (2009). Simultane liniennetz-und taktfahrlagenoptimierung. *Der Eisenbahningenieur*, *60*(6), 6–10.

Rodríguez-Martín, I. & Salazar-González, J. J. (2010). A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research*, *37*(3), 575–581.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.

Saharidis, G. K. & Ierapetritou, M. G. (2013). Speed-up benders decomposition using maximum density cut (mdc) generation. *Annals of Operations Research*, *210*(1), 101–123.

Şahin, G. & Yüceoğlu, B. (2011). Tactical crew planning in railways. *Transportation Research Part E: Logistics and Transportation Review*, *47*(6), 1221–1243.

Schmid, V. & Ehmke, J. F. (2015). Integrated timetabling and vehicle scheduling with balanced departure times. *OR spectrum*, *37*(4), 903–928.

Schmidt, M. & Schöbel, A. (2015). Timetabling with passenger routing. *OR spectrum*, *37*(1), 75–97.

Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR spectrum*, *34*(3), 491–510.

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

Schöbel, A. & Scholl, S. (2006). Line planning with minimal traveling time. In Kroon, L. G. & Möhring, R. H. (Eds.), *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*, volume 2 of *OpenAc-*

cess Series in Informatics (OASIcs), Dagstuhl, Germany.

Schrage, L. (2018). *A Guide to Optimization-Based Multiperiod Planning*, chapter 3, (pp. 50–63). INFORMS.

Serafini, P. & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, *2*(4), 550–581.

Sharp, G., Jones, P., & Bell, J. (1974). Application of a route structuring algorithm to city bus lines.

Siebert, M. & Goerigk, M. (2013). An experimental comparison of periodic timetabling models. *Computers & Operations Research*, *40*(10), 2251–2259.

Silman, L. A., Barzily, Z., & Passy, U. (1974). Planning the route system for urban buses. *Computers & operations research*, *1*(2), 201–211.

Smet, P., Wauters, T., Mihaylov, M., & Berghe, G. V. (2014). The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega*, *46*, 64–73.

Torres, L. M., Ramiro, T., Borndörfer, R., & Pfetsch, M. E. (2011). Line planning on tree networks with applications to the quito trolebús system. *International Transactions in Operational Research*, *18*(4), 455–472.

Torres, L. M., Torres, R., Borndörfer, R., & Pfetsch, M. E. (2008). Line planning on paths and tree networks with applications to the quito trolebus system. In Fischetti, M. & Widmayer, P. (Eds.), *8th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'08)*, volume 9 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany.

Tsekeris, T. & Tsekeris, C. (2011). Demand forecasting in transport: Overview and modeling advances. *Economic research-Ekonomska istraživanja*, *24*(1), 82–94.

Vaidyanathan, B., Jha, K. C., & Ahuja, R. K. (2007). Multicommodity network flow approach to the railroad crew-scheduling problem. *IBM Journal of Research and Development*, *51*(3.4), 325–344.

van Nes, R., Hamerslag, R., & Immers, L. (1988). *The design of public transport networks*, volume 1202. National Research Council, Transportation Research Board.

Vanderbeck, F. & Savelsbergh, M. W. (2006). A generic view of dantzig–wolfe decomposition in mixed integer programming. *Operations Research Letters*, *34*(3), 296–306.

Wang, J., Zhou, L., & Yue, Y. (2019). Column generation accelerated algorithm and optimisation for a high-speed railway train timetabling problem. *Symmetry*, *11*(8), 983.

W.Lampkin & P.D.Saalmans (1967). The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *Journal of the Operational Research Society*, *18*(4), 375–397.

Yang, R., Han, B., Zhang, Q., Han, Z., & Long, Y. (2022). Integrated optimization of train route plan and timetable with dynamic demand for the urban rail transit line. *Transportmetrica B: Transport Dynamics*, 1–34.

Yue, Y., Wang, S., Zhou, L., Tong, L., & Saat, M. R. (2016). Optimizing train stopping patterns and schedules for high-speed passenger rail corridors. *Transportation Research Part C: Emerging Technologies*, *63*, 126–146.

Zäpfel, G. & Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics*, *113*(2), 980–996.

Zhang, C., Qi, J., Gao, Y., Yang, L., Gao, Z., & Meng, F. (2021). Integrated optimization of line planning and train timetabling in railway corridors with passengers' expected departure time interval. *Computers & Industrial Engineering*, *162*, 107680.

Zhou, X. & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B: Methodological*, *41*(3), 320–341.