Large-Scale Analytics and Optimization in Urban Transportation: Improving Public Transit and Its Integration with Vehicle-Sharing Services

by

Virot Chiraphadhanakul

B.Eng., Chulalongkorn University (2007)
S.M., Massachusetts Institute of Technology (2010)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author . . . .
Sloan School of Management
May 17, 2013

Certified by . . . .
Cynthia Barnhart
Ford Professor of Civil and Environmental Engineering
Associate Dean, School of Engineering
Thesis Supervisor

Accepted by . . . .
Patrick Jaillet
Dugald C. Jackson Professor, Department of Electrical Engineering
and Computer Science
Co-Director, Operations Research Center

Large-Scale Analytics and Optimization in Urban Transportation: Improving Public Transit and Its Integration with Vehicle-Sharing Services

by

Virot Chiraphadhanakul

Submitted to the Sloan School of Management
on May 17, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Public transportation is undeniably an effective way to move a large number of people in a city. Its ineffectiveness, such as long travel times, poor coverage, and lack of direct services, however, makes it unappealing to many commuters. In this thesis, we address some of the shortcomings and propose solutions for making public transportation more preferable.

The first part of this thesis is focused on improving existing bus services to provide higher levels of service. We propose an optimization model to determine limited-stop service to be operated in parallel with local service to maximize total user welfare. Theoretical properties of the model are established and used to develop an efficient solution approach. We present numerical results obtained using real-world data and demonstrate the benefits of limited-stop services.

The second part of this thesis concerns the design of integrated vehicle-sharing and public transportation services. One-way vehicle-sharing services can provide better access to existing public transportation and additional options for trips beyond those provided by public transit. The contributions of this part are twofold. First, we present a framework for evaluating the impacts of integrating one-way vehicle-sharing service with existing public transportation. Using publicly available data, we construct a graph representing a multi-modal transportation service. Various evaluation metrics based on centrality indices are proposed. Additionally, we introduce the notion of a transfer tree and develop a visualization tool that enables us to easily compare commuting patterns from different origins. The framework is applied to assess the impact of Hubway (a bike-sharing service) on public transportation service in the Boston metropolitan area. Second, we present an optimization model to select a subset of locations at which installing vehicle-sharing stations minimizes overall travel time over the integrated network. Benders decomposition is used to tackle large instances. While a tight formulation generally generates stronger Benders cuts, it requires a large number of variables and constraints, and hence, more computational effort. We propose new algorithms that produce strong Benders cuts quickly by aggregating various variables and constraints. Using data from the Boston metropolitan area, we present computational experiments that confirm the effectiveness of our solution approach.

Thesis Supervisor: Cynthia Barnhart
Title: Ford Professor of Civil and Environmental Engineering
Associate Dean, School of Engineering

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Cynthia Barnhart for everything she has done for me. Her expertise, guidance, and advices have contributed greatly to this thesis, and I would not have come this far without her constant support and encouragement. Her kindness and caring have made my journey at MIT a very pleasant one. She has been an exceptional role model for me, and it is a great honor to have known her. Working with her in the past six years has helped me grow both professionally and personally.

I would like to thank Prof. Amedeo Odoni and Prof. Patrick Jaillet, who served on my thesis committee, for their suggestions and insightful comments on this thesis, and more importantly, for their kind help since my day one at MIT as an MST student. I also feel very fortunate to have worked closely with Amedeo as a teaching assistant for the 1.200 course. He is a truly amazing teacher.

Additionally, this thesis has benefited from discussions with a number of people: Prof. Nigel Wilson, Prof. Haris Koutsopoulos, Prof. Carolina Osorio, Prof. Robert Hampshire, Kent Larson, Dr. Ryan Chin, Dimitris Papanikolaou, Stephen Zoepf, and Scott Mullen. I would also like to acknowledge the sponsors of my research work, Ford Motor Company and the National Research Foundation of Singapore through the Singapore-MIT Alliance for Research and Technology's Future Urban Mobility research program. I am very grateful to Maria Marangiello, Laura Rose, and Andrew Carvalho for their administrative assistance. Special thanks go to Maria for always trying very hard to find me a good time slot from Cindy's tight schedule.

Many thanks go to my friends at the ORC who have made it a vibrant community. Special, special thanks go to Lavanya Marla, Vikrant Vaze, Matthieu Monsch, Cristian Figueroa, and Joline Uichanco, whose friendship made my stay at MIT much more enjoyable and memorable. Importantly, I would like to thank all Thai friends in Boston and Cambridge areas for creating a home-like atmosphere for me.

I am also indebted to all teachers in my life, without whom I would not be where I am today. My special appreciation goes to Prof. Manoj Lohatepanont and Prof.

Proadpran Punyabukkana for supporting me in this endeavor.

Finally, I owe my deepest gratitude to my family for their love and support. They are my everything, and I dedicate this work to them. Thanks especially to Noey for always being there.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The world's urban population grew rapidly over the last century. According to a UN report (United Nations Population Fund, 2007), more than half of the world's population, 3.3 billion, lived in urban areas in 2008, and the number of urbanites is expected to swell to almost five billion by 2030. As a result of urbanization, the increasing mobility demands in many cities have exceeded the available transportation capacity, causing severe traffic congestion and other consequences. The Texas Transportation Institute estimates the cost of traffic congestion in U.S. urban areas in 2010 to be as high as $101 billion, which results from 4.8 billion hours of travel delay and 1.9 billion gallons of wasted fuel (Lomax et al., 2011).

Public transportation is undeniably an effective way, economically and environmentally, to move a large number of people in a city. It reduces traffic congestion, fuel consumption, and carbon footprint. An average single-occupancy vehicle emits 76% more greenhouse gas per passenger mile than heavy rail transit (subways, metros, or rapid transit) and 33% more than buses (Hodges, 2010). In 2010, the presence of public transportation in 439 U.S. urban areas saved 300 million gallons of fuel and 796 million hours of delay, amounting to $16.8 billion of cost savings (Lomax et al., 2011). In addition to the social impacts, an American can save, on average, $9,917 annually by switching from owning and driving private vehicles to riding public transportation, according to the American Public Transportation Association's Transit Savings Report (August, 2012).

Despite the benefits of public transportation, a number of people still prefer driving their private vehicles. Almost 80% of Americans drove alone to work in 2010 (U.S. Census Bureau). Some common reasons that discourage people from taking public transportation are poor coverage and accessibility, inflexible and infrequent schedules, unreliable services, lack of direct services (hence, multiple transfers required), long travel times, cargo carrying needs, privacy, and safety concerns.

In this thesis, we address some of these shortcomings and propose solutions for making public transportation more preferable.

1.2 Contributions and Thesis Outline

The first part of this thesis is focused on improving existing bus services to provide higher levels of service. One of the major disadvantages of bus service, making it an unattractive choice for many commuters, is long in-vehicle travel times resulting from frequent stops. Limited-stop bus services have the advantage of shorter in-vehicle times for passengers and shorter running times that enable bus operators to serve more demand with the same number of buses and reduced operating costs. In Chapter 2, we seek to find an optimal way to introduce a limited-stop service to be operated in parallel with an existing local bus service. We propose an optimization model to determine: (1) the bus stops along a route to be served by a limited-stop service; and (2) the frequencies of the limited-stop and the local services that maximize total user welfare, for a given route during a given time period. A few theoretical properties of the model are established and used to develop a solution approach. As a proof of concept, we present numerical results obtained using real-world data together with comprehensive discussions of solution quality, computational times and the model's sensitivity to different parameters. Moreover, we solve the optimization model for 178 real-world bus routes with different characteristics in order to demonstrate the impacts of some key attributes on the potential benefits of limited-stop services.

The second part of this thesis concerns the design of integrated vehicle-sharing and public transportation services. Vehicle-sharing service (or short-term vehicle rental service) has been growing tremendously in the past two decades and becoming an attractive alternative for urban transportation. It provides flexible, personal mobility without the full cost of vehicle ownership. Importantly, in the case of one-way vehicle-sharing service, where users do not need to drop-off a vehicle at the pick-up station, it also complements existing public transportation systems. In particular, with a proper design of a one-way system, it can provide better access to existing public transportation and additional options for trips beyond those provided by public transit.

In spite of the many benefits of public transportation, it is economically infeasible to provide high levels of public transportation service over scattered urban areas. Additionally, the environmental benefits of public transit are only materialized when capacities are utilized to a significant extent. Therefore, the potential impacts of integrating one-way vehicle-sharing service with existing public transportation can be far-reaching.

In Chapter 3, we present a framework for evaluating the impacts of integrating one-way vehicle-sharing service with existing public transportation. Using publicly available data and web services, we model a graph representing a multi-modal transportation service. Various metrics based on centrality indices are proposed. Additionally, we introduce the notion of a transfer tree that details the transfer hierarchy of trips originating from a given transit node. A web-based interactive visualization tool is developed and used to compare commuting patterns from different origins and identify critical transit hubs that are not well connected by the network under consideration. As a case study, we apply the framework to assess the impact of Hubway, a bike-sharing program in the Boston metro area, on the existing public transportation system operated by the Massachusetts Bay Transportation Authority (MBTA).

In Chapter 4, we present an optimization model to select a subset of locations at which installing vehicle-sharing stations minimizes overall travel time over the integrated vehicle-sharing and public transportation network. Benders decomposition is used to tackle large instances. While a tight formulation generally generates stronger Benders cuts, it involves a large number of variables and constraints, and hence, more computational effort. We propose new algorithms that produce strong Benders cuts quickly by aggregating various variables and constraints. Using data from the Boston metropolitan area, we present computational experiments that confirm the effectiveness of our solution approach.

Finally, we conclude this thesis in Chapter 5.

Chapter 2

Incremental Bus Service Design: Combining Limited-Stop and Local Bus Services

2.1 Introduction

One of the major disadvantages of bus service, making it an unattractive choice for many commuters, is long in-vehicle travel times resulting from frequent stops. Limited-stop bus services on high-demand corridors, such as those successfully operated in cities such as Bogota, Chicago, Montreal, New York City, and Santiago, however, have the advantage of shorter in-vehicle times for passengers and shorter running times that enable bus operators to serve more demand with the same number of buses and reduced operating costs.

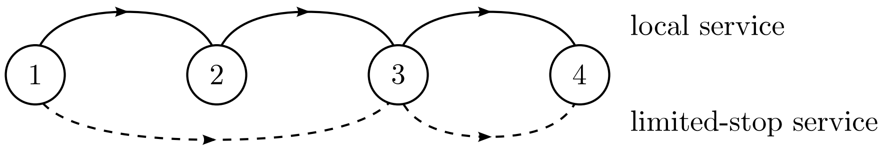


Figure 2-1: An example of a 4-stop corridor with a local route overlapping with a limited-stop route serving stops 1, 3, and 4.

Given the benefits of limited-stop bus service, we are interested in finding an optimal way to introduce a limited-stop service to be operated in parallel with an existing local bus service, which serves every bus stop along the corridor (see Figure 2-1). In this work, we focus on incremental changes to the existing schedule. In particular, we seek to modify a given bus service on a particular corridor by optimally reassigning some number of (local) bus trips, as opposed to providing additional trips, to operate a limited-stop service. This ensures that the new operation does not require additional buses and incurs no extra cost. Additionally, we consider introducing only one additional limited-stop route to facilitate adoption of the new service and avoid complicated operations.

The challenges in the incremental bus service design problem are as follows. There are trade-offs between in-vehicle time reduction and out-of-vehicle time increase. Specifically, while the passengers served by a limited-stop service experience shorter in-vehicle travel times, those served only by the reduced frequency local service have to wait longer for a bus. Additionally, we need to take into account passenger behavior in response to a limited-stop service. For passengers that are served by both local and limited-stop service, some may board the first bus to arrive; while the others may wait for the limited-stop service. A passenger's choice depends on the travel time savings he/she can get from the limited-stop service.

One of the major goals of this work is to develop a tractable optimization model together with an efficient solution approach that can be used to solve the incremental bus service design problem for real-world bus services. We propose an optimization model to determine: (1) the bus stops along a route to be served by a limited-stop service; and (2) the frequencies of the limited-stop and the local services that maximize total user welfare, for a given route during a given time period. A few theoretical properties of the model are established and used to develop a solution approach. Despite the implementation of limited-stop services around the world and the extensive literature on bus network design, there are, to our knowledge, only two published works, one by Leiva et al. (2010) and the other by Ulusoy et al. (2010) that develop optimization models for the design of limited-stop services overlapping with local services. The major differences between earlier published works and our approach are summarized in Table 1, and discussed in detail in what follows. Using data from a bus operator in a major city, we present numerical results as a proof of concept. Through the results, we also examine solution quality, computational times and the model's sensitivity to different parameters.

Another goal of this work is to provide insights into limited-stop bus service design. Solving the optimization model for 178 bus services with different characteristics, we investigate the impacts of some key attributes discussed in Scorcia (2010), Leiva et al. (2010), and Larrain et al. (2010) on potential benefits of limited-stop services.

The chapter is organized as follows. In the next section, we review the literature related to the design of limited-stop service. In Section 2.3, we describe the incremental bus service design problem and present the optimization model. In Section 2.4, we present the solution approach consisting of three key parts: decomposition, problem size reduction, and a heuristic. The numerical results together with discussions on solution quality, computational times and sensitivity analyses are provided in Section 2.5. In Section 2.6, we discuss how some characteristics of a bus service impact the potential benefits of limited-stop services. Finally, in Section 2.7, we conclude and suggest future research on this topic.

2.2 Literature Review

Despite the extensive literature on public transit network design (see Ceder and Wilson, 1986; Desaulniers and Hickman, 2007; Ceder, 2007; Guihaire and Hao, 2008), there are, to our knowledge, only two published works, one by Leiva et al. (2010) and the other by Ulusoy et al. (2010) that develop optimization models for the design of limited-stop services overlapping with local services.

Leiva et al. (2010) formulate a mixed integer nonlinear model to determine frequencies of a set of *predefined* limited-stop routes such that the social costs, comprising user and operator costs, are minimized. For passenger assignment, they assume that every passenger chooses a set of acceptable lines based on the expected total

	Leiva et al. (2010)	Ulusoy et al. (2010)	This work
Assumptions:			
- O-D matrix	Fixed	Fixed	Fixed
- Transfers	Allowed	Allowed	Not Allowed
- Number of limited-stop routes allowed	Unlimited	Unlimited	1
Objective	Minimize social costs	Minimize social costs	Maximize user welfare
Constraints:			
- Capacity	✓(heuristic)	✓	✓
- Fleet size	-	✓	✓
Passenger Assignment	Proportional to the frequencies of each attractive line	A logit-based model considering wait, transfer, and in-vehicle times	A linear function of frequency share and in-vehicle travel time savings
Solution Approach	*Leiva et al. (2010)*: Given a set of predefined limited-stop routes, find optimal frequencies using a nonlinear program without capacity constraints. Iteratively increase the frequencies of overcrowded lines until the capacity constraints are satisfied. *Ulusoy et al. (2010)*: Exhaustively search over all predefined limited-stop routes and all possible frequencies for an optimal solution to a mixed integer nonlinear model. *This work*: For each frequency allocation, find an optimal limited-stop route using a mixed integer program together with an algorithm for reducing problem size. Then, select the frequency allocation that yields the highest objective value. A heuristic is proposed to further improve computational times		

Table 2.1: Comparison of papers by Leiva et al. (2010), Ulusoy et al. (2010), and this work

travel times and always takes the first bus to come. Consequently, demand for each acceptable line is proportional to its frequency and does *not* depend on its travel time savings. In order to obtain an optimal solution, they first limit the complexity of the model by omitting the capacity constraints and solve the resulting nonlinear model (no binary decision variables). If the optimal solution violates the capacity constraint, a heuristic is then applied to progressively increase the frequencies of the overcrowded lines until the capacity constraints are satisfied. They present numerical results for a bus service along a 19-stop corridor with 23 predefined limited-stop routes (including express services, short turning, and deadheading). Computational times however are not provided. Additionally, they examine the impacts of different demand profiles on the objective function value. This topic is further discussed in Larrain et al. (2010).

One major advantage of this work is the flexibility of the optimization model, which allows transfers, multiple limited-stop routes along a corridor, and non-homogeneous fleet (big and small buses). Nevertheless, the numerical results show that transfer do not occur if the transfer penalties are high, and the additional benefits from having more than two limited-stop routes operated along a corridor is minimal.

Similarly, Ulusoy et al. (2010) formulate a mixed integer nonlinear model to determine frequencies of a set of *predefined* limited-stop routes such that the social costs are minimized. The main advantage of this work is that they estimate demand for each service according to the in-vehicle, wait, and transfer times using a logit-based model. This however results in an intractable nonlinear model. They propose an exhaustive search algorithm for obtaining an optimal set of frequencies. Using data from a real-world rail transit line, they present numerical results for a 6-station service. Computational times are again not provided.

In addition to the papers discussed above, Scorcia (2010) extends the work of Schwarcz (2004) and proposes a comprehensive framework for the design and evaluation of limited-stop and BRT services overlapping with local services. The model evaluates limited-stop service configurations based on six measures of effectiveness including demand split between local and limited-stop services and change in average passenger travel time. Although the work does not involve any optimization, they are particularly useful for evaluating limited-stop services obtained from optimization models.

Finally, there is another set of closely related works that provide optimization models and algorithms for other bus route planning strategies for high-demand corridors such as deadheading (Ceder and Stern, 1981; Furth, 1985), short turning (Ceder, 1989; Furth, 1987) and zonal service (Jordan and Turnquist, 1979; Furth, 1986).

2.3 Incremental Bus Service Design

Given the benefits of limited-stop bus services, we seek to modify a given bus schedule on a particular corridor by optimally reassigning some number of bus trips, as opposed to providing additional trips, to operate a limited-stop service in parallel with the local service, which serves every stop along the corridor. Consequently, the new operation does not require additional buses and incurs no extra costs. In this section, we present an optimization model to determine, for a given bus route, (1) the bus stops along the route to be served by a limited-stop service; and (2) the frequencies of the limited-stop and the local services that maximize total user welfare. It is important to note that we consider introducing only one additional limited-stop route, as implemented in many cities, to facilitate adoption of the new service and avoid complicated operations. Moreover, Leiva et al. (2010) find that additional benefits of having more than two limited-stop routes operated along a corridor are minimal.

2.3.1 Basic Assumptions

In the proposed model, we assume the following.

1. The O-D demand is given and fixed. Specifically, we assume that passengers will continue to board and alight at the stops they previously prefer and not walk to nearby stops that are served by both local and limited-stop services. However, the demand split between the local and limited-stop services is captured in the model and determined according to the attractiveness of services, demand

elasticity, and available capacities.

2. Passengers arrive randomly at their origins at a constant rate over the time period under consideration.

3. Passengers are assigned on each bus service according to a system-optimal assignment, rather than user-optimal assignment. The validity of this assumption will be discussed further in Section 2.5.2.

4. Transfers between local and limited-stop services are not allowed. The validity of this assumption depends on the cost of a transfer. Leiva et al. (2010) show in their case study that transfers do not occur with high transfer penalties.

2.3.2 Model Formulation

Consider an existing bus service serving bus stops in a set $S = \{1, 2, \dots, |S|\}$. The bus route begins at stop 1 and ends at stop $|S|$. Operated by a homogeneous fleet of buses with capacity of C passengers, the service currently runs at a constant frequency[1] of f_0 trips over a period under consideration (e.g., AM peak or PM peak) of length T minutes. Let K denote the set of origin-destination (O-D) pairs served by this bus service, which is given by $\{k = (s^k, d^k) \mid s^k, d^k \in S, s^k < d^k\}$. The expected demand for an O-D pair $k \in K$ over the time period is p^k passengers. Lastly, an expected travel time saving from running express from stop i to stop $j (j > i)$ (i.e., a service stops at stops i and j and skips every stop between i and j) is c_{ij} minutes. Note that if stops i and j are adjacent, that is $j = i + 1$, the expected travel time saving c_{ij} equals zero.

Decision Variables

The decision variables used in the model are summarized as follows. The first set of variables is related to limited-stop service, and the second set is related to passenger assignment.

[1] Note that the term 'frequency' in this work refers to *the number of bus trips* operated over a period under consideration of length T minutes

f = number of limited-stop bus service trips over the period under consideration

α_{ij} = 1 if the limited-stop service runs express from stop i to stop j (i.e., stops consecutively at stop i and j), where $i, j \in S$ and $j > i$

β_i = 1 if the limited-stop service serves stop $i \in S$

γ^k = 1 if the limited-stop service serves O-D pair $k \in K$. Specifically, an O-D pair $k = (s^k, d^k) \in K$ is served by a limited-stop service if and only if both origin s^k and destination d^k are served by the limited-stop service.

x_{ij}^k = portion of passengers of O-D pair $k \in K$ assigned to the express segment from stops i to j ($i, j \in S$ and $j > i$) of limited-stop service

y^k = portion of passengers of O-D pair $k \in K$ assigned to local service

z^k = portion of passengers of O-D pair $k \in K$ preferring the limited-stop service

w_i = number of passengers on the local service traveling from stops i to $i + 1$

Objective Function

The objective of our model is to maximize total user welfare, which is defined as total in-vehicle time savings for the passengers served by the limited-stop service, minus the total increase in wait time, weighted by disutility of wait time relative to in-vehicle time μ_w, for those served by the reduced frequency local service and those preferring the limited-stop service. Because we fix the total number of bus trips operated by local and limited-stop services, the new operation incurs no extra cost, and we omit operator cost from the objective function. Mathematically, the objective function is given by

$$\begin{aligned}\text{Maximize} \qquad & \sum_{k \in K} p^k \sum_{(i,j) \in \Gamma^k} c_{ij} x_{ij}^k \\ & - \mu_w \sum_{k \in K} p^k (1 - \gamma^k) \frac{1}{2} \left(\frac{T}{f_0 - f} - \frac{T}{f_0} \right) \\ & - \mu_w \sum_{k \in K} p^k z^k \frac{1}{2} \left(\frac{T}{f} - \frac{T}{f_0} \right),\end{aligned} \tag{2.1}$$

where Γ^k denotes a set of segments that can be used to serve O-D pair $k \in K$. Mathematically, for an O-D pair $k = (s^k, d^k) \in K$, Γ^k is given by $\{(i,j) \mid i,j \in S, s^k \leq i < j \leq d^k\}$.

The first term is the total in-vehicle time savings for the passengers served by the limited-stop service. For an O-D pair served only by local service, no passengers can be assigned on the limited-stop service (i.e., x_{ij}^k's are zero), and hence this term is zero. The second term corresponds to the total increase in the expected wait time (in equivalent in-vehicle minutes) for passengers served by the reduced frequency local service. For an O-D pair served by both local and limited-stop services, we have that $1 - \gamma^k$ equals 0, and this term is zero. Finally, the last term represents the total increase in the expected wait time (in equivalent in-vehicle minutes) for passengers preferring the limited-stop service.

Note that in Equation (2.1), we have the expected wait times equal half the headway, assuming vehicle arrivals are equally spaced with perfect headway. In general, for a random headway H, the expected waiting time for a randomly arriving passenger is equal to $\frac{\sigma_H^2 + E^2[H]}{2E[H]}$, where $E[H]$ and σ_H^2 are the mean and variance of the headway H. Therefore, the perfect headway assumption can be relaxed by replacing a factor of $\frac{1}{2}$ with an appropriate value. Additionally, this form of wait time function may not be appropriate for low-frequency service, for which passengers tend to time their arrivals according to the published schedule. Nonetheless, because we focus on incremental changes to high-frequency service, operating every 15 minutes or less, passengers are presumably accustomed to not timing their arrivals and will continue this practice even after the limited-stop service is introduced.

Limited-Stop Service Route Constraints

In order to allow a limited-stop service route to begin and end at any bus stops in S, we introduce dummy stops s^+ and s^- at which a limited-stop service route virtually begins and ends, respectively. The following set of constraints ensures that the values of α_{ij}'s constitute a valid route.

$$\sum_{i \in S \setminus \{|S|\}} \alpha_{s^+,i} = 1 \tag{2.2}$$

$$\sum_{j \in S: j<i} \alpha_{ji} + \alpha_{s^+,i} = \sum_{j \in S: j>i} \alpha_{ij} + \alpha_{i,s^-} \qquad \forall i \in S \tag{2.3}$$

$$\sum_{i \in S \setminus \{1\}} \alpha_{i,s^-} = 1 \tag{2.4}$$

$$\alpha_{ij} \in \{0,1\} \qquad \forall (i,j) \in \{(i,j) \mid i,j \in S, i<j\} \tag{2.5}$$

$$\alpha_{s^+,i}, \alpha_{i,s^-} \in \{0,1\} \qquad \forall i \in S$$

Note that a limited-stop route serving exactly one stop (i.e., all α_{ij}'s are zeros except $\alpha_{s^+,i}$ and α_{i,s^-} for some $i \in S$) is also valid according to constraints (2.2)-(2.5). Such routes however cannot serve any passengers, while increasing wait times of all passengers. Therefore, their corresponding objective function values are negative, and they cannot be an optimal solution.

Given the values of α_{ij}'s, the values of β_i's and γ^k's can then be obtained through the following constraints.

$$\beta_i = \sum_{j \in S: j>i} \alpha_{ij} + \alpha_{i,s^-} \qquad \forall i \in S \tag{2.6}$$

$$\gamma^k \leq \beta_{s^k} \qquad \forall k = (s^k, d^k) \in K \tag{2.7}$$

$$\gamma^k \leq \beta_{d^k} \qquad \forall k = (s^k, d^k) \in K \tag{2.8}$$

$$\beta_i \in \{0,1\} \qquad \forall i \in S \tag{2.9}$$

$$\gamma^k \in \{0,1\} \qquad \forall k \in K \tag{2.10}$$

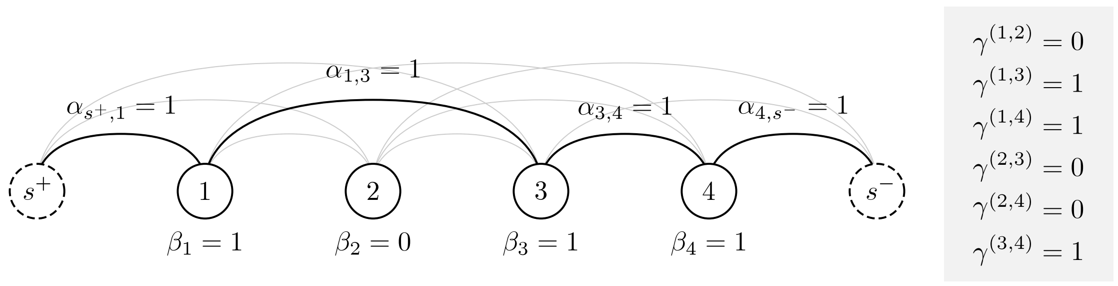


Figure 2-2: An example of a 4-stop corridor with a limited-stop route serving stops 1, 3, and 4.

Note that a stop $i \in S$ is served by a limited-stop service ($\beta_i = 1$) if there exists an express segment starting at stop i in the limited-stop service route. According to constraints (2.7) and (2.8), for each O-D pair $k = (s^k, d^k) \in K$, if both origin and destination are served by the limited-stop service ($\beta_{s^k} = \beta_{d^k} = 1$), the value of γ^k in an optimal solution must be 1 in order to maximize the objective function value. Figure 2-2 illustrates the values of α_{ij}'s, β_i's and γ^k's for a 4-stop corridor with a limited-stop service serving stops 1, 3, and 4.

Passenger Flow Constraints

The validity of passenger flows is captured by the following constraints.

$$x_{ij}^k \le \alpha_{ij} \qquad \forall k \in K, (i,j) \in \Gamma^k \tag{2.11}$$

$$y^k + \sum_{j \in S: s^k < j \le d^k} x_{s^k,j}^k = 1 \qquad \forall k = (s^k, d^k) \in K \tag{2.12}$$

$$\sum_{j \in S: s^k \le j < i} x_{ji}^k - \sum_{j \in S: i < j \le d^k} x_{ij}^k = 0 \qquad \begin{array}{l} \forall k = (s^k, d^k) \in K, \\ i \in \{i \in S \mid s^k < i < d^k\} \end{array} \tag{2.13}$$

$$0 \le x_{ij}^k \le 1 \qquad \forall k \in K, (i,j) \in \Gamma^k \tag{2.14}$$

$$0 \le y^k \le 1 \qquad \forall k \in K \tag{2.15}$$

In words, constraints (2.11) ensure that each passenger can be assigned on an express segment only if the segment is included in the limited-stop service route. Imposed by constraint (2.12), the model assigns every passenger to either local or

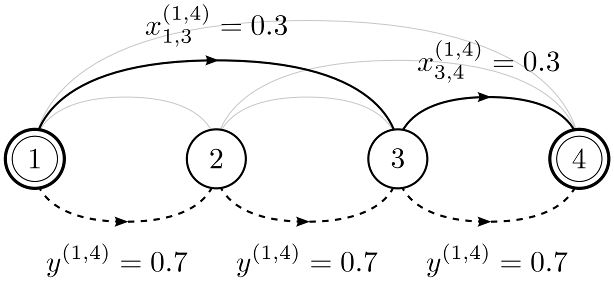
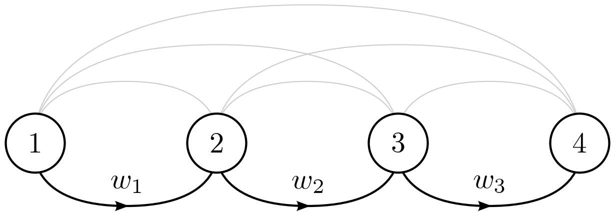


Figure 2-3: Example flows of passengers on O-D pair $(1, 4)$ on both local service and limited-stop service serving stops 1, 3, and 4.

limited-stop service. Additionally, for a given O-D pair $k = (s^k, d^k) \in K$, the flow of passengers on the limited-stop service is conserved at each stop between s^k and d^k by constraint (2.13). Figure 2-3 depicts flows of passengers on O-D pair $(1, 4)$ (i.e., traveling from stop 1 to 4) on both local service ($y^{(1,4)}$) and limited-stop service ($x_{ij}^{(1,4)}$'s).

Capacity Constraints

The total number of passengers on each service cannot exceed its total capacity, defined as frequency multiplied by bus capacity. Note that although the total capacity of both services is greater than the given travel demand, this set of constraints is still needed to ensure that the numbers of limited-stop service passengers and local service passengers each do not exceed their respective capacities. While the number of passengers on each segment of the limited-stop service route can be computed directly from the x_{ij}^k's, the number of passengers on each segment of the local service has to be derived through the w_i's (see Figure 2-4). Thus, the capacity constraints



$$w_2 = p^{(1,3)}y^{(1,3)} + p^{(1,4)}y^{(1,4)} + p^{(2,3)}y^{(2,3)} + p^{(2,4)}y^{(2,4)}$$

Figure 2-4: Number of passengers on the local service traveling between adjacent stops.

can be written as follows.

$$w_1 = \sum_{\substack{k \in K, \\ s^k = 1}} p^k y^k \tag{2.16}$$

$$w_i = \sum_{\substack{k \in K, \\ s^k \leq i}} p^k y^k - \sum_{\substack{k \in K, \\ d^k \leq i}} p^k y^k \qquad \forall i \in S \setminus \{1, |S|\} \tag{2.17}$$

$$0 \leq w_i \leq (f_0 - f)C \qquad \forall i \in S \setminus \{|S|\} \tag{2.18}$$

$$\sum_{k \in K : \Gamma^k \ni (i,j)} p^k x_{ij}^k \leq (fC)\alpha_{ij} \qquad \forall (i,j) \in \{(i,j) \mid i,j \in S, i < j\} \tag{2.19}$$

$$f \in \{1, 2, \ldots, f_0 - 1\} \tag{2.20}$$

Because our definition of frequency is the number of bus trips operated over a period under consideration, we are only interested in integral values of frequency. Additionally, we can ignore the cases where f equals 0, and f equals f_0. Clearly, when f is zero, the solution yields an objective function value of zero as there is no change to the original operation. When f equals f_0, that is, all buses operate a limited-stop service, the limited-stop service route must contain every stop along the corridor; otherwise, some O-D demand would not be satisfied. As a result, the limited-stop service becomes the local service, and again, there is no change to the original operation.

Demand Split Constraints

One of the challenges for this optimization problem formulation is to capture passenger behavioral changes in response to a new limited-stop service. We model demand for a limited-stop service as follows:

$$\sum_{j \in S : s^k < j \le d^k} x^k_{s^k,j} \le \frac{f}{f_0} + a^k \left(\sum_{(i,j) \in \Gamma^k} \alpha_{ij} c_{ij} \right) \qquad \forall k \in K, \tag{2.21}$$

where a^k's are constants representing the incremental proportion of passengers preferring the limited-stop service per minute of travel time reduction.

From the equation, the demand for a limited-stop service is a linear function of frequency share and travel time reduction, which is given by the term in parentheses. If a limited-stop service does not provide any travel time reduction, passengers are indifferent between the local and limited-stop services and board the first bus to arrive. In this case, the demand for the limited-stop service is proportional to its frequency relative to the local service. As the reduction in travel time increases, the demand for the limited-stop service increases linearly at the rate of a^k because some passengers are willing to wait longer for the limited-stop service, as opposed to boarding the first arriving bus. One possible choice of a^k, which we use in this work, is the negative of travel time elasticity divided by the expected travel time of O-D pair k on the local service.

We refer to the portion of passengers assigned to a limited-stop service beyond its frequency share ($\frac{f}{f_0}$) as those *preferring* the limited-stop service. Mathematically, the portion of passengers on O-D pair k *preferring* the limited-stop service (z^k) can be obtained through the following constraints.

$$z^k \ge \sum_{j \in S : s^k < j \le d^k} x^k_{s^k,j} - \frac{f}{f_0} \qquad \forall k \in K \tag{2.22}$$

$$0 \le z^k \le 1 \qquad \forall k \in K \tag{2.23}$$

Because the objective function improves as z^k decreases, the constraints ensure that z^k is equal to $\max\Big(0, \sum_{j \in S: s^k < j \le d^k} x^k_{s^k,j} - \frac{f}{f_0}\Big)$ in an optimal solution.

Lastly, note that if an O-D pair $k \in K$ is not served by a limited-stop service, x^k_{ij}'s must be zero, and hence the constraint (2.21) is redundant.

We close this section by establishing one important property of the model:

Proposition 1. *The integrality of β_i's and γ^k's can be relaxed.*

Proof. Because constraints (2.2)-(2.4) together with (2.5) ensure that the right hand side of constraint (2.6) is either 0 or 1, we can simply omit the integrality constraint (2.9) of β_i's. Now consider the value of γ^k associated with O-D pair $k = (s^k, d^k)$. From constraints (2.7) and (2.8), if β_{s^k} and/or β_{d^k} take the value 0, γ^k must also be 0—an integral value. If both β_{s^k} and β_{d^k} equal 1, without the integrality constraint (2.10), γ^k may take any real value from 0 to 1, while all the constraints are still satisfied. In the optimal solution, however, γ^k has to take the value 1 in order to maximize the objective function, provided that p^k is positive. If p^k is 0, then γ^k can take any value without affecting the optimal solution. □

2.4 Solution Approach

In this section, we present a solution approach to the mixed integer nonlinear model described earlier. The solution approach, consisting of three key parts, allows us to solve the incremental bus service design problem for real-world bus services efficiently.

2.4.1 Decomposition

Note that the nonlinearity in our model is caused by the limited-stop service frequency variable f in the capacity constraint (2.19) and the objective function (2.1). If a value of f is fixed, the model will become linear and can be solved for an optimal limited-stop service route more easily. We therefore decompose the original optimization problem into two stages. First, we repeatedly solve the optimization model assuming different limited-stop service frequencies. Then, given the set of optimal limited-stop routes for different limited-stop service frequencies, we select the limited-stop service frequency that yields a limited-stop route with the highest objective function value. This can be done because we are interested in values of f from a finite set $\{1, 2, \ldots, f_0 - 1\}$. In particular, let z be the optimal objective function value of the optimization model in Section 2.3 and $z(f)$ be the optimal objective function value for a fixed f, we have that

$$z = \underset{f \in \{1,2,\ldots,f_0-1\}}{\text{Maximize}} \, z(f).$$

When it is optimal to have no limited-stop service, the optimal limited-stop service routes for every f in $\{1, 2, \ldots, f_0 - 1\}$ will be identical to local service, yielding an objective function value of zero.

One might attempt to establish a systematic way to search for the optimal limited-stop route frequency. However, we empirically found that $z(f)$ is not necessarily a unimodal function of f, and therefore search algorithms might find only a local optimum. This occurs because of the discrete nature of the limited-stop service route decisions. Consequently, we have to exhaustively search over the set of possible values of f in oder to ensure optimality. Nevertheless, there are usually a small number of possible frequency allocations for a period under consideration (e.g., AM peak or PM peak), and only frequency shares within a certain range are likely to be of interest to transit agencies.

2.4.2 Problem Size Reduction

The mixed-integer linear model resulting from fixing the value of limited-stop service frequency remains difficult to solve for large instances, that is, problems with many stops along the routes. It has some parallels to the facility location problem, which is commonly known as a hard problem. One possible way to limit computational complexity is to reduce problem size. To do so, we derive upper bounds on the contributions of α_{ij}'s to the objective function value, and then use these to eliminate some variables.

Recall that α_{ij} is equal to 1 when a limited-stop service serves stops i and j and no other stops in between, i.e., runs express from stops i to j. Thus, every passenger whose origin or destination is between stops i and j is not served by the limited-stop service and will experience increased expected wait time of $\delta_f = \frac{1}{2}\left(\frac{T}{f_0-f} - \frac{T}{f_0}\right)$. On the other hand, those whose trips start before stop i *and* end after stop j *might* benefit from the in-vehicle time savings of c_{ij} minutes on the limited-stop service. The actual contribution to the objective function is subject to available capacity, the demand split between local and limited-stop services, and whether their origins and destinations are served by limited-stop service. By assuming that every stop before i and after j is served by limited-stop service, the *maximum possible* in-vehicle time savings from running express from stops i to j is given by $c_{ij}\min\left(fC, \sum_{k\in K:\Gamma^k\ni(i,j)} p^k\right)$. Therefore, an upper bound on the contribution of α_{ij} to the objective function value, for a given limited-stop frequency f, is given by

$$U_{ij}(f) = c_{ij}\min\left(fC, \sum_{k\in K:\Gamma^k\ni(i,j)} p^k\right) - \mu_w \sum_{\substack{\{k\in K|i<s^k<j\}\\ \cup\{k\in K|i<d^k<j\}}} p^k\delta_f. \tag{2.24}$$

For any pair of stops i and j, a variable α_{ij} can then be eliminated from the formulation if the upper bound $U_{ij}(f)$ is negative for a given limited-stop service frequency f. Because variables x_{ij}^k's for all $k \in K$ such that $(i,j) \in \Gamma^k$ can take positive values only when α_{ij} equals 1, the corresponding x_{ij}^k's can also be eliminated.

Additionally, we observe and prove the following property of this upper bound.

Proposition 2. *For $0 \le f \le f' < f_0$, if $U_{ij}(f) < 0$, then $U_{ij}(f') < 0$.*

Proof. We first claim that the upper bound $U_{ij}(f)$ is a concave function of $f \in [0, f_0)$. From equation (2.24), the two terms inside the minimum operator are linear, and hence concave, in f. Because concavity is preserved under minimum operation, we have that the first term in equation (2.24) is concave. The second term (including the minus sign) is also concave as its second derivative is non-positive for $f \in [0, f_0)$. Lastly, because concavity is preserved under summation, we have that $U_{ij}(f)$ is a concave function of $f \in [0, f_0)$. Next, note that $U_{ij}(0) = 0$. Depending on the first derivative of $U_{ij}(\cdot)$ at 0, the value of $U_{ij}(f)$ may increase as f increases until the first derivative becomes zero. The concavity ensures that once the value of $U_{ij}(f)$ falls below 0 for some f, it remains negative for all $f' > f$. We thus establish the claim. □

In other words, once a variable α_{ij} is eliminated for some f, it will also be eliminated for any $f' > f$. More importantly, the property suggests that it is generally easier to solve the optimization model for a large limited-stop service frequency f as more variables are likely to be eliminated.

2.4.3 Heuristic

The underlying idea of this heuristic arises from the observation that the optimal limited-stop service route for a particular limited-stop service frequency f is almost identical to the optimal limited-stop service routes for $f - 1$. Potentially, an optimal solution for a particular limited-stop service frequency f can be valuable input to the optimization model for $f - 1$ to reduce computational complexity.

Note that for a large limited-stop service frequency f, the optimal limited-stop service route tends to skip only a few stops because of the limited capacity of the local service and the substantial increase in wait time for passengers who are not served by the limited-stop service. As a limited-stop service becomes less frequent, the local service capacity increases; the increase in wait time for local passengers diminishes; and consequently, an optimal limited-stop route tends to skip more stops. Empirically, we observe that if a certain stop is not included in the optimal limited-stop route for a limited-stop service frequency f, that stop is also not included in the optimal routes for any $f' < f$. Thus, we propose the following heuristic that can be used to solve a sequence of optimization problems for different limited-stop service frequencies.

1. Solve the optimization model for a limited-stop service frequency $f = (f_0 - 1)$. Let $\beta_i^*(f)$ denote the optimal values of β_i's for a limited-stop service frequency f.

2. For each stop i in S, if $\beta_i^*(f)$ is zero, add the constraint $\beta_i = 0$ to the optimization model.

3. Solve the optimization model with the additional constraints for a limited-stop service frequency $f = f - 1$.

4. Repeat steps 2 and 3 for less frequent limited-stop service.

We start the heuristic with a large limited-stop service frequency because it is easier to solve as suggested by the property in Proposition 2.

Finally, although the optimality of the solutions obtained from this heuristic has not been proved analytically, we have not found any instances where this heuristic leads to suboptimal solutions.

2.5 Proof of Concept

Using data from a bus operator in a major city, we provide in this section numerical results obtained from our optimization model and solution approach. We also examine solution quality, computational times, and the model's sensitivity to different parameters.

2.5.1 Data and Parameters

We obtain real-world data from a bus operator in a major city. The data set contains route information and expected O-D demands of 178 high-frequency bus routes, operating every 15 minutes or less. In this work, we focus on the weekday, morning peak schedules, from 7 a.m. to 9 a.m. ($T = 120$ minutes).

Table 2.2 summarizes the baseline values and expressions of the parameters we use in this work. We assume that the travel times between adjacent stops are equal for simplicity, and more importantly, to facilitate our understanding of the model behavior and solutions. The dwell times at each stop are also assumed to be equal for the same reasons. We acknowledge that instead of constants, dwell times should be a

Parameter		Baseline Value
Wait time disutility weight	μ_w	1.0
Bus capacity	C	80 passengers
Travel time between adjacent stops	$t_{i,i+1}$	1.5 minutes
Dwell time at stop i	t_i^d	0.5 minutes
Travel time elasticity	e	-0.5
Total travel time from stop i to stop j on a local service	t_{ij}	$\sum_{l=i}^{j-1} t_{l,l+1} + \sum_{l=i+1}^{j-1} t_l^d$
Total travel time savings from running express between stops i and j	c_{ij}	$\sum_{l=i+1}^{j-1} t_l^d$
Rate of increase in limited-stop service demand per minute of travel time reduction for O-D pair $k = (s^k, d^k)$	a^k	$\frac{-e}{t_{s^k,d^k}}$

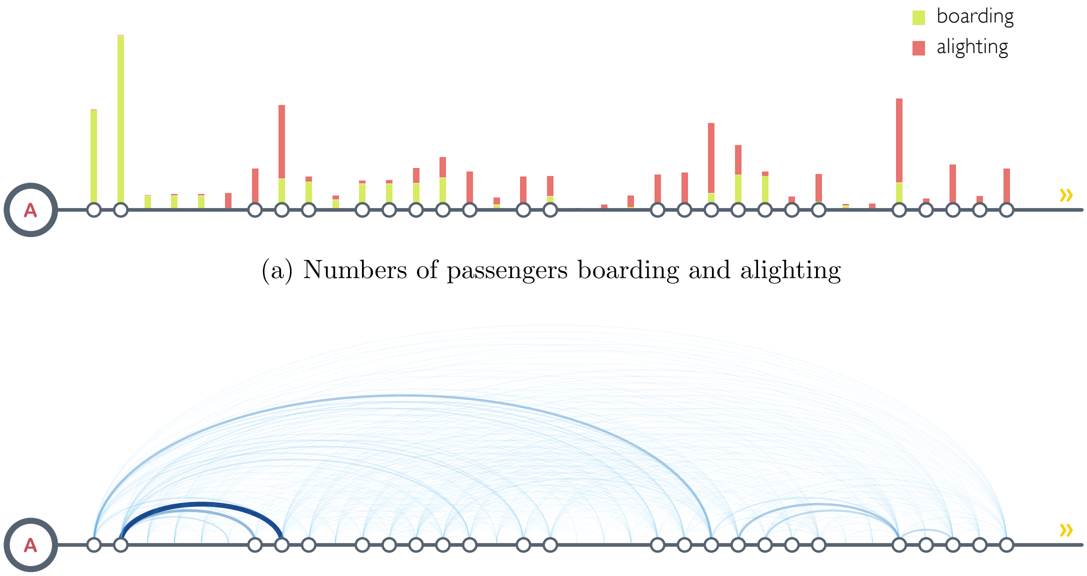
Table 2.2: Baseline values of parameters.

function of the expected numbers of passengers boarding and alighting at the stops, which in turn depend on the decision variables—limited-stop service route, frequency allocation, and the resulting passenger assignments. Incorporating variable dwell times however will result in an intractable model. Lastly, we simply specify the total travel time savings from running express between two stops as the sum of the dwell times at the skipped stops (i.e., every stop between the two stops). To be more precise, rigorous estimates of limited-stop service run times like the one presented in Tétreault and El-Geneidy (2010) can be used.

In this section, our discussions focus on one particular bus service, referred to as bus service A. Bus service A consists of 35 stops along its corridor spanning almost 9 miles. With 24 bus trips, the service carries on average a total of 3,151 passengers during the two-hour morning peak period. The average trip length of passengers served by service route A is 11 stops.

2.5.2 Numerical Results

We implemented the solution approach proposed in Section 2.4 with Java 1.6 and IBM ILOG CPLEX 12.2. The result obtained from the optimization model shows



(a) Numbers of passengers boarding and alighting

(b) O-D Demands

Figure 2-5: Visualization of the optimal limited-stop service route together with (a) numbers of passengers boarding and alighting at each stop and (b) O-D demands.

that, by reassigning 13 out of 24 bus trips (54%) to operate a limited-stop service, a total user welfare of 1,506 minutes can be achieved on route A. Covering 84% of the demand, the limited-stop route skips 11 out of 35 stops (31%) along the corridor. Figure 2-5 visualizes the optimal limited-stop service route together with (a) numbers of passengers boarding and alighting at each stop and (b) O-D demands. A hollow circle represents a bus stop which is served by the limited-stop route. A height of a bar at each stop in Figure 2-5a indicates the number of passengers boarding/alighting at the stop. Thickness and opacity of an arc connecting two stops in Figure 2-5b indicates a proportion of passengers on the O-D pair.

It is evident that most bus stops with high demands for boarding and/or alighting are included in the limited-stop service route as the service can then potentially benefit a large number of passengers. However, the bus stops that are served by the limited-stop service do not necessarily have higher demands than those skipped stops. For example, stop 6, which is not served by the limited-stop service, has slightly higher demand than stop 32 (the fourth from last), which is served by the limited-stop service. This is because stopping at stop 6 affects in-vehicle travel times of a

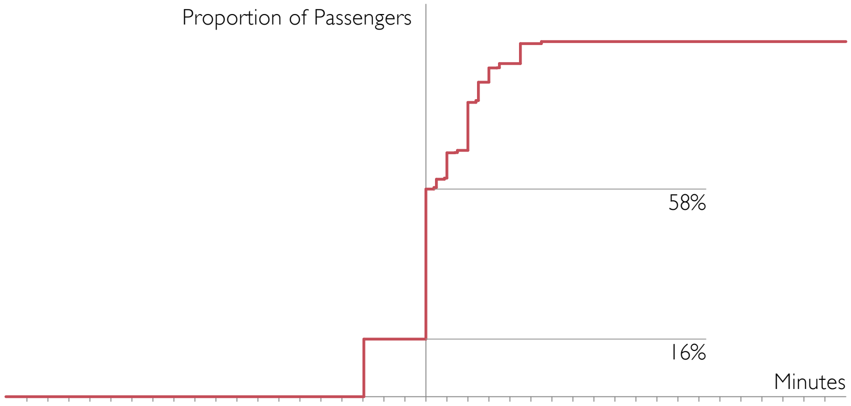


Figure 2-6: Cumulative distribution function (CDF) of travel time changes.

large number of passengers boarding the limited-stop service at the first and second stops. Additionally, there are very few passengers boarding at stop 6, and those who alight at stop 6 gain only little benefit from the limited-stop service. On the other hand, although there are fewer passengers alighting at stop 32, a lot of them travel longer distances, thereby contributing larger in-vehicle travel time reduction to the objective function value. Moreover, stopping at stop 32 affects in-vehicle travel times of a smaller number of passengers on the limited-stop service who alight at the last three stops.

It is essential to understand how passenger travel times (wait and in-vehicle times) change compared to the original service. Figure 2-6 shows a cumulative distribution of the travel time changes. About 16% of the passengers are not served by the limited-stop service and have to wait on average 3 minutes longer for the frequency-reduced local service. Another 42% of the passengers are not affected by the introduction of the limited-stop service. In particular, these passengers are served by both local and limited-stop services, but the limited-stop service does not provide any travel time reductions to their trips (i.e., every stop between their origins and destinations is served by the limited-stop service). Lastly, the other 42% of the passengers benefit from travel time reductions ranging from 0.4 to 5.5 minutes. The average travel time reduction is 2.3 minutes.

The distribution of the travel time changes suggests that passengers are not likely to make transfers between local and limited-stop service because besides the inconvenience of transferring and the possibility of additional fare costs, the potential travel time savings are offset for most passengers by additional wait times at transfer points. Specifically, if a passenger first boards the local service and connects to the limited-stop service, the additional wait time is $\frac{1}{2}(\frac{120}{13}) = 4.6$ minutes on average; and if a passenger first boards the limited-stop service and connects to the local service, the additional wait time is $\frac{1}{2}(\frac{120}{11}) = 5.5$ minutes on average. Therefore, for this particular route, our assumption that transfers between local and limited-stop services are not allowed is generally valid. Similarly for the other 177 high-frequency bus routes in our network, we also find that, in each optimal solution, the average wait time for both local and limited-stop services is larger than the average travel time reduction provided by the limited-stop service.

In terms of ridership split, 55% and 45% of the passengers are assigned on the local and limited-stop services, respectively. Figure 2-7 visualizes a bus's load profile for each service. The dashed line at the top indicates the bus capacity. The limited-stop service is generally less crowded than the local service. Moreover, it is important to note that the limited-stop buses are never full, which implies that the capacity constraints of the limited-stop service are not tight. Consequently, all the passengers who want to get on a limited-stop bus (either because it is the first bus to arrive, or because he/she *prefers* the limited-stop service) can board, and the system-optimal assignment is identical to the user-optimal assignment. This is also the case for the other bus routes in our network. Nonetheless, in general, when a limited-stop service capacity is reached—especially in subproblems with small limited-stop service frequencies, the optimal solution corresponds to the system-optimal assignment, and its objective function value serves as an upper bound on the total user welfare in the user equilibrium.

2.5.3 Computational Times

One major goal of this work is to develop an optimization model together with a solution approach that can be used to solve efficiently the incremental bus service

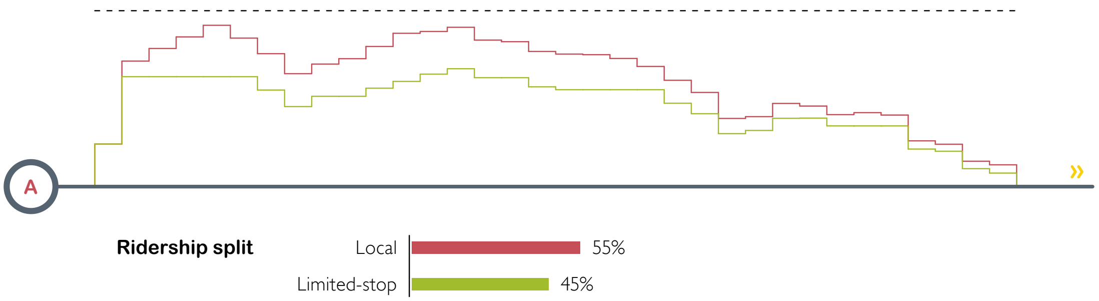


Figure 2-7: Load profiles of the local and limited-stop services.

design problem for real-world bus services. We present in this section computational times and discussions on the effectiveness of the solution approach proposed earlier.

Computations are carried out on a Mac OS X machine with an Intel Core i7 2.7 GHz processor and 8 GB of RAM. We decompose the problem into 23 subproblems, one for each possible value of limited-stop service frequency f. The computational time for each subproblem is limited to 300 seconds. Table 2.3 summarizes the computational times for bus service A when the problem size reduction and/or the heuristic presented in Section 2.4 are applied. In order to compute the optimality gap of a solution, we obtain the optimal solution for each subproblem by applying the problem size reduction and letting the CPLEX MIP solver run without imposing a time limit.

As a baseline, we first solve each subproblem using only the MIP solver, that is, neither the problem size reduction nor the heuristic presented in Section 2.4 are applied. For f less than or equal to 11, the solver cannot obtain any feasible solution within the time allotted; and for f between 12 and 14, the solutions provided by the solver are not optimal.

The next set of computational times are obtained by using the upper bound in (2.24) to reduce the problem sizes before running the MIP solver. Although feasible solutions still cannot be obtained within the time limit for f less than or equal to 11, the computational times are significantly reduced for the other values of f. Moreover, the optimal solutions for f equal to 13 and 14 can now be obtained within the time limit.

For the third set of computational times, we apply the heuristic outlined in Section

Limited-stop service frequency	Computational time in seconds (Non-zero Optimality Gap)						
	None		Problem size reduction only		Heuristic only		Both
1	-		-		63.81	(19%)	298.93
2	-		-		299.03	(20%)	299.07
3	-		-		298.96		299.10
4	-		-		299.10		298.88
5	-		-		225.30		137.35
6	-		-		144.66		86.10
7	-		-		113.87		64.62
8	-		-		72.03		40.58
9	-		-		38.73		28.12
10	-		-		22.83		13.86
11	-		-		14.65		5.87
12	289.38	(35%)	297.49	(13%)	12.72		4.29
13	289.52	(31%)	297.79		8.85		2.20
14	289.25	(6%)	128.12		4.49		1.09
15	266.14		40.84		3.65		0.65
16	139.66		23.16		11.78		1.94
17	94.83		8.92		23.12		2.03
18	46.83		4.36		13.54		1.73
19	36.49		1.48		20.76		0.63
20	22.00		0.60		5.49		0.20
21	7.72		0.16		5.57		0.09
22	4.29		0.05		2.36		0.04
23	1.30		0.02		1.44		0.02

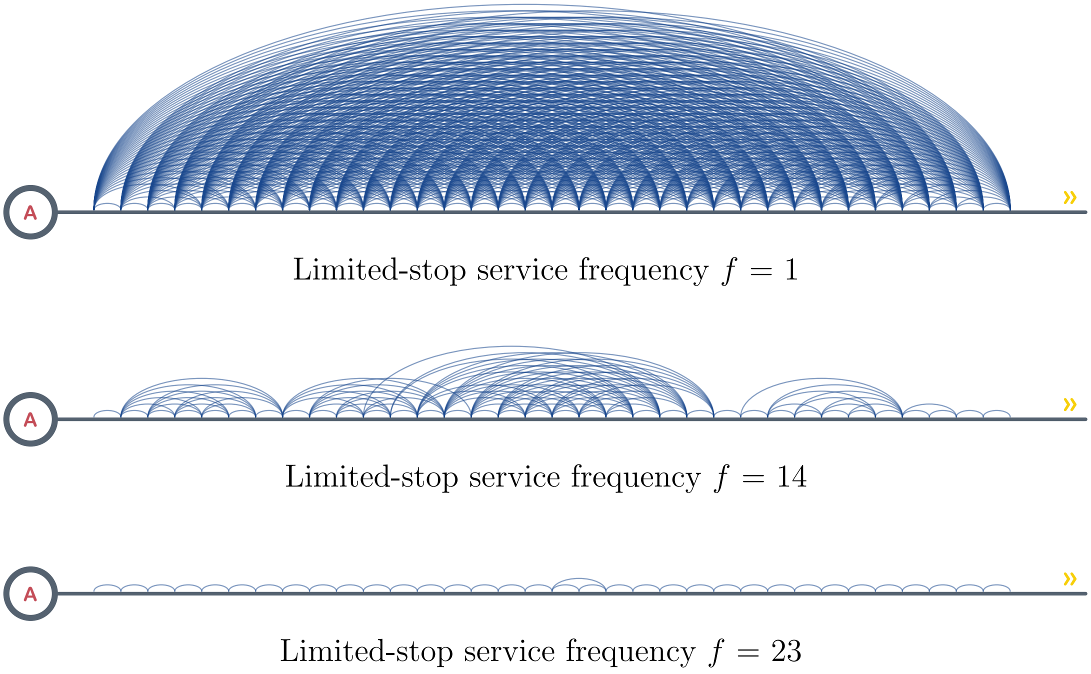
Table 2.3: Computational time.

2.4.3 without the problem size reduction. The heuristic enables us to obtain the optimal solutions within the time limit for all possible limited-stop service frequencies except for f equal to 1 and 2, where suboptimal feasible solutions are obtained. Note that for f equal to 1, the MIP solver stops before the time limit is reached with a nonzero optimality gap. This occurs because the heuristic adds more constraints to the original formulation according to the solution for f equal to 2, which is not optimal, and consequently leads the MIP solver to an incorrect optimal solution. The computational times are again significantly reduced from the baseline, although the problem size reduction appears to be more effective for f greater than or equal to 17.

Finally, we apply both the problem size reduction and the heuristic. The optimal solutions can now be obtained within the time limit for all possible limited-stop service frequencies, and the computational times are further reduced.

Effectiveness of problem size reduction using the upper bound in (2.24) is depicted in Figure 2-8. In the figure, each arc connecting stops i and j represents a variable α_{ij} that has a nonnegative upper bound on the contribution to the objective function value and hence remains in the optimization model. For a limited-stop frequency of 1, only about 2% of variables α_{ij}'s are eliminated. As implied by Proposition 2—more variables can be eliminated for higher limited-stop service frequencies, 80% and 94% of variables α_{ij}'s are eliminated for a limited-stop service frequencies of 14 and 23, respectively. For the latter, there are only two possible routes from the remaining variables—one serves every stop (like the local service) and the other serves every stop except stop 19. Therefore, the optimization model can be solved extremely quickly.

Moreover, the optimal limited-stop service routes for different limited-stop service frequencies in Figure 2-9 illustrates the optimality of the heuristic for bus service A. In particular, it can be seen that when a stop is skipped in an optimal limited-stop route for a limited-stop service frequency f, the stop is also skipped in the optimal limited-stop routes for any limited-stop service frequencies smaller than f.



Limited-stop service frequency $f = 1$

Limited-stop service frequency $f = 14$

Limited-stop service frequency $f = 23$

Figure 2-8: Remaining variables after problem size reduction using the upper bound in (2.24).

2.5.4 Sensitivity Analyses

The numerical results presented in Section 2.5.2 are obtained using the baseline values of parameters. In this section, we examine how changes in the values of parameters affect the total user welfare for bus service A. The results are summarized in Figure 2-10. A solid circle in each plot indicates the baseline case. Additionally, the characteristics of limited-stop service in the optimal solution of each scenario are provided in Table 2.4.

Dwell Time. Dwell times depend on many factors such as traffic conditions, passenger loads, fare payment methods, busway designs, and vehicles (floor heights, door configurations). Increasing the dwell time at a stop causes increases in total in-vehicle travel times for services serving the stop and total travel time savings of limited-stop services skipping the stop. Because, in this work, the total travel time savings is simply defined as a sum of dwell times at the skipped stops, varying dwell times essentially adjusts travel time savings achievable by limited-stop services. As a

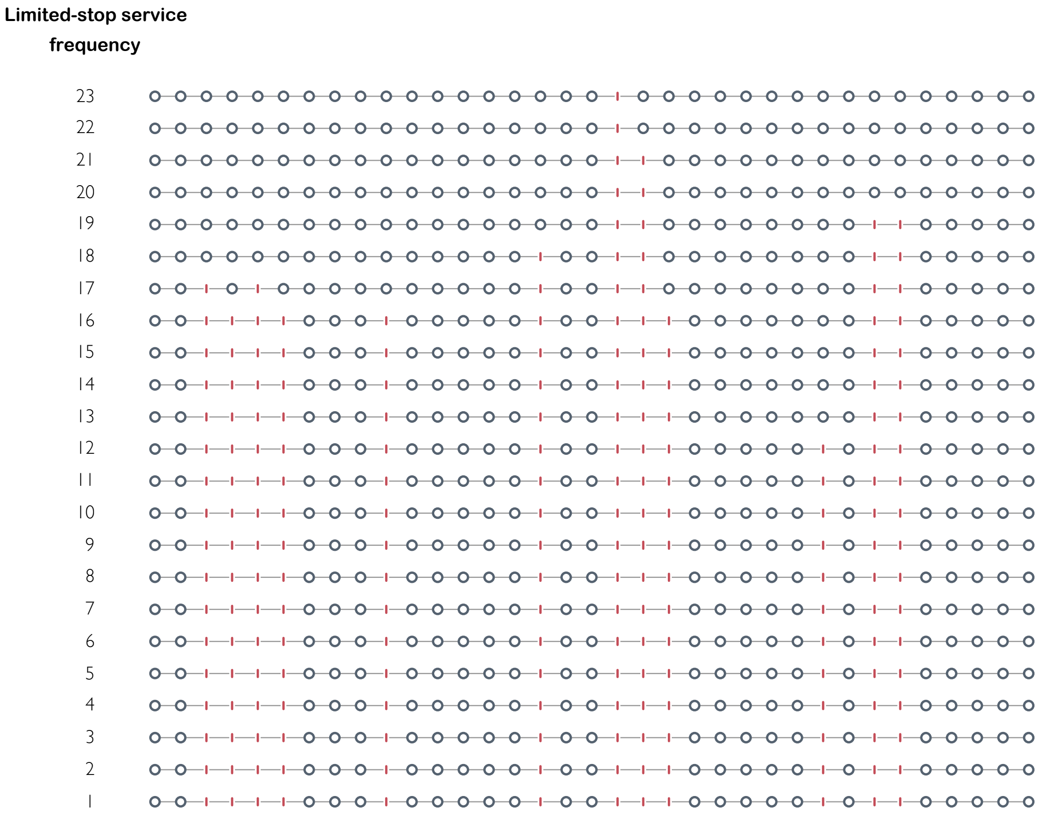
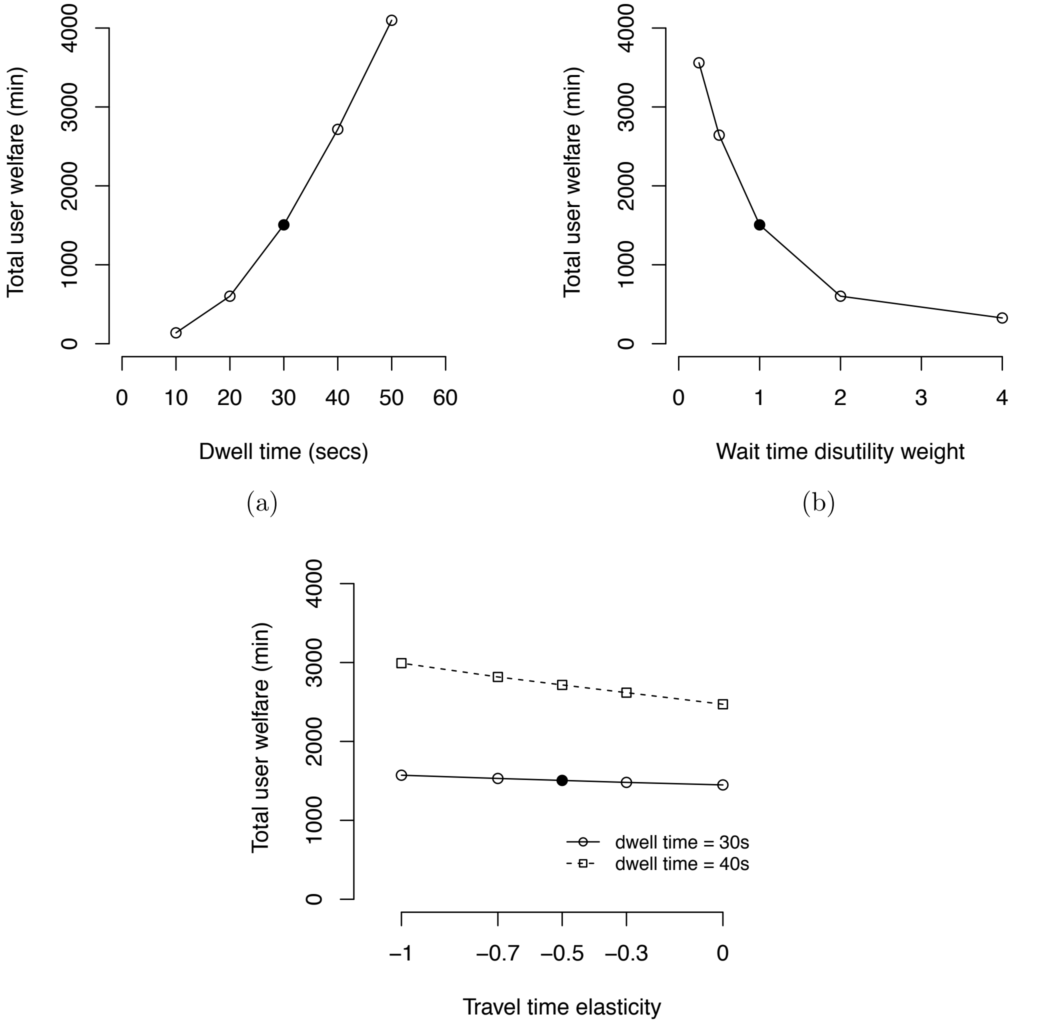


Figure 2-9: Optimal limited-stop service routes for different limited-stop service frequencies.



(a)

(b)

(c)

Figure 2-10: Sensitivity analyses.

	Total user welfare	Limited-stop service		
	(in-vehicle minutes)	Frequency	No. of stops skipped	Ridership (%)
Dwell time (secs)				
10	138	14	2	43
20	602	10	10	33
30	1,506	13	11	45
40	2,716	14	12	52
50	4,098	15	12	56
Wait time disutility weight				
0.25	3,560	17	13	58
0.5	2,643	16	12	57
1	1,506	13	11	45
2	602	12	5	37
4	326	12	2	39
Travel time elasticity	(dwell time = 30s)			
-1.0	1,572	14	11	55
-0.7	1,531	13	11	45
-0.5	1,506	13	11	45
-0.3	1,481	13	11	42
0.0	1,449	12	11	39
Travel time elasticity	(dwell time = 40s)			
-1.0	2,992	15	12	60
-0.7	2,818	15	12	57
-0.5	2,716	14	12	52
-0.3	2,618	14	12	50
0.0	2,471	14	11	44

Table 2.4: Optimal solutions from sensitivity analyses.

result, total user welfare increases as the dwell time per stop increases. Additionally, Figure 2-10a shows that these increases are more substantial as dwell times increase.

In-vehicle : wait time disutility. So far, we assume that one minute of in-vehicle travel time poses the same level of disutility as one minute of wait time. In the transportation literature, wait time cost is usually assumed to be larger than in-vehicle time cost primarily because of discomfort caused by weather, safety, etc. On the other hand, transit agencies in many cities around the world now provide real-time bus arrival information, allowing passengers to schedule their arrivals at bus stops such that their wait times are minimized, regardless of the scheduled headway of the bus service. In this case, the cost of wait time amounts to the cost of schedule delay, which might be less than or equal to the in-vehicle time cost as passengers can remain productive until the next bus arrives.

According to Figure 2-10b, as wait time disutility increases, the total user welfare drops at a decreasing rate. This suggests that reassigning some number of local bus trips to operate a limited-stop service is particularly beneficial to bus systems for which the wait time cost or disutility is relatively low compared to the in-vehicle time cost.

Travel time elasticity. Travel time elasticity usually varies from one city to another and depends on many factors such as income, trip length and time of day. Typically, travel time elasticity ranges between -0.3 and -0.7. According to constraints (2.21)-(2.23), increasing travel time elasticity can potentially increase the proportion of passengers *preferring* limited-stop service. The increase is however subject to travel time reductions relative to travel times on the local service and the additional wait time resulting from not boarding the first arriving bus. For dwell times of 30 seconds, total user welfare decreases minimally as travel time elasticity increases (see Figure 2-10c). This happens because the travel time reductions achievable by limited-stop services are small relative to the travel times on the local service. Specifically, only 1.4% of passengers *prefer* the limited-stop service in the baseline case. We also perform a sensitivity analysis for dwell times of 40 seconds to see the effect of travel time elasticity when the travel time reductions achievable by limited-stop services are higher. In this case, the changes in total user welfare are more significant. Lastly, note that because we assume a fixed O-D matrix, travel time elasticity only affects total user welfare through the ridership split between the local and limited-stop services. In reality, it also affects the number of new riders attracted to the limited-stop service.

2.6 Insights into Limited-Stop Bus Service Design

The tractability of the optimization model together with our efficient solution approach allows us to solve the incremental bus service design problem for all 178 high-frequency bus routes in our data set, in which the longest bus route consists of 104 stops along its corridor spanning 23.4 miles. Given the optimal solutions for these bus services with different characteristics, we examine the impacts of some key attributes discussed in Scorcia (2010), Leiva et al. (2010), and Larrain et al. (2010) on the potential benefits of limited-stop services.

The attributes of bus services we consider in this work are demand volume (passengers), service frequency (buses), route length (stops), average trip length (stops), and demand variability (dimensionless). For a loop service, because once a bus completes its service in one direction, it continues the service in the reverse direction right away, we treat the service in both directions as a continuous service on one long corridor, and hence the route length is given by the total number bus stops along both directions. Additionally, we measure demand variability using the coefficient of variation of the total demand (boarding and alighting) at each bus stop. In order to fairly compare the benefits of limited-stop services among different bus services, we calculate the total user welfare attained by an optimal solution as a percentage of the total travel time, including both wait and in-vehicle travel time, of all passengers.

Figure 2-11 summarizes the statistics and correlations of different attributes and total user welfare. Specifically, the diagonal panels show the distributions of each attribute in our data set. The upper diagonal panels are scatter plots for each pair of attributes, and their associated correlation coefficients are provided in the lower diagonal panels.

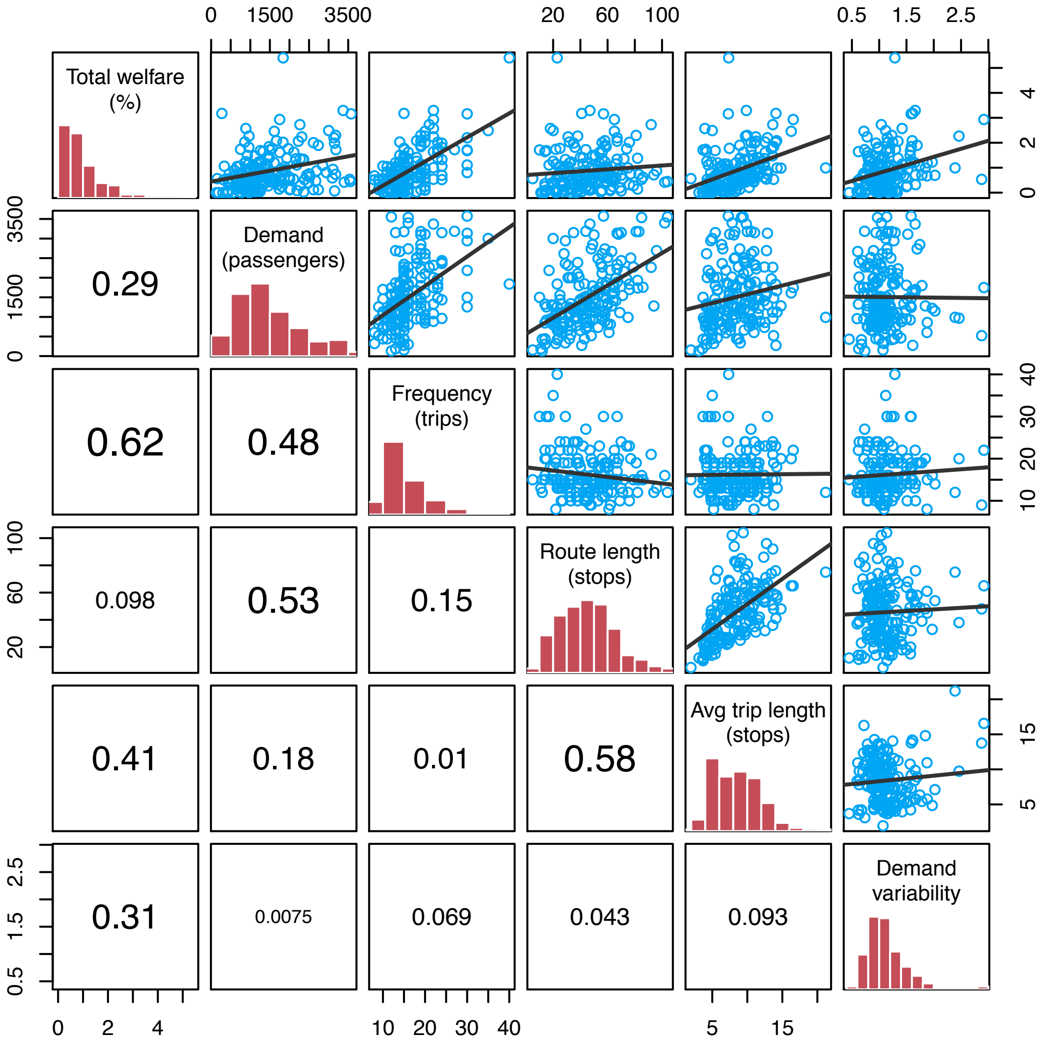


Figure 2-11: Statistics and correlations of different bus service attributes and total user welfare.

Service frequency has the highest correlation with total user welfare. This is reasonable because if an original service frequency is low, reassigning some bus trips to operate a limited-stop service will drastically increase wait times for those who are only served by the reduced frequency local service. Although it is commonly known that limited-stop services are promising for high-demand corridors, the correlation between demand and total user welfare is not particularly high. One possible reason is that, despite the high demand, many passengers may only make short trips, thereby not gaining large benefits from limited-stop service. Nevertheless, because high demand generally implies a large number of passengers who can potentially benefit from a limited-stop service, the correlation coefficient between the demand and the total user welfare *in minutes*, as opposed to the relative percentage, is as high as 0.56.

With the second highest correlation with total user welfare, average trip length is another key attribute that determines the benefits of limited-stop services. For every O-D pair, it takes into account both the number of passengers and how much they can potentially benefit from limited-stop services. Even though route length is highly correlated with average trip length, it is barely correlated with total user welfare, and hence not an accurate indicator of a successful limited-stop service. Again, this is simply because passengers do not necessarily travel along the entire long route. However, because a long bus route typically serves more passengers, the correlation coefficient between the route length and the *absolute* total user welfare (in minutes) is as high as 0.34.

Lastly, demand variability exhibits correlation with total user welfare to some extent. High demand variability corresponds to a concentration of demands, for both boarding and alighting, at certain stops. Consequently, high demand variability allows a limited-stop service to serve a number of passengers without making frequent stops and therefore increases the potential benefits of limited-stop services. Recall that we measure demand variability using the coefficient of variation of the total demand at each stop. As a result, it does not capture how high-demand stops are distributed along the corridor. In particular, if the high-demand stops are close together, having a limited-stop service serving all the stops will result in minimal travel time reduction, while having a limited-stop service skipping some of the stops will increase wait times of many passengers. In short, high demand variability can only partially indicate the demand patterns or load profiles that allow for successful limited-stop services. Additionally, Scorcia (2010) points out that it is important to have some minimal level of demand at low-demand stops. Otherwise, local service may spend short amounts of time at the stops or even skip them often, and hence, limited-stops services provide little additional travel time savings. Because in this work, we assume a constant dwell time at each stop, our results cannot demonstrate this observation.

2.7 Conclusions and Future Work

This work addresses the incremental bus service design problem, in which we seek to modify a given bus service by optimally reassigning some number of bus trips to operate a limited-stop service without incurring extra operating costs. We formulate a mixed integer nonlinear model to determine the limited-stop service to be operated in parallel with the local service, and to optimize its associated frequency to maximize total user welfare. Exploiting some theoretical properties of the model, the proposed solution approach consists of three parts: decomposition, problem size reduction, and a heuristic. Although the optimality of the heuristic has not been proved analytically, we have not found any instances where the heuristic leads to suboptimal solutions.

Using real-world data from a bus operator in a major city, as a proof of concept, we provide numerical results together with detailed discussion regarding solution quality, including the distribution of travel time changes and ridership split between local and limited-stop services. The reported computational times demonstrate the tractability of the model and effectiveness of the solution approach. The sensitivity analyses shows that

- as travel time savings achievable by limited-stop services increases, the benefits of limited-stop services, measured by total user welfare, increase at an increasing rate;

- reassigning some number of local bus trips to operate a limited-stop service is particularly beneficial to the bus systems for which the wait time cost is relatively low compared to the in-vehicle time cost; and

- the impact of travel time elasticity is noticeable only when the travel time reductions achievable by limited-stop services are large relative to the travel times on the local service.

Moreover, we solve the optimization model for 178 bus routes with different characteristics in order to examine the impacts of some key attributes on the potential benefits of limited-stop services. We find that service frequency and average trip length are highly correlated with the total user welfare attained by the optimal solutions, while demand volume and route length show reasonable correlation with total user welfare only in an absolute sense. Lastly, demand variability exhibits correlation with total user welfare only to some extent as it still depends on the underlying demand profile.

As in most mathematical models in the public transit network design literature, we make certain assumptions to simplify the problem and ensure tractability of the model. In terms of future research, the following are particularly interesting directions:

- *Relaxing the fixed O-D matrix assumption.* This assumption prohibits passengers whose origins or destinations are not served by a given limited-stop service from walking to nearby stops that are served by both local and limited-stop services. The assumption can be restrictive if the travel time savings from a limited-stop service are considerable and sufficiently justify walking to other bus stops. Additionally, this ignores potential ridership increase in the long run.

- *Considering multiple bus routes that share a segment along their corridors simultaneously.* This is important because introducing a limited-stop service for one bus route will not only affect the ridership split between the local and limited-stop services, but might also lead to ridership shift from one bus route to another.

- *Taking into account shorter running times of limited-stop service.* As a result of shorter running times of limited-stop service, an operator might be able to operate more bus trips over a period under consideration using the same number of vehicles. This however will increase the operating costs, and the objective function can no longer omit the operator cost to ensure profitability.

- *Incorporating limited-stop service frequency into the demand model.* In the presented model, the portion of passengers preferring a limited-stop service only depends on the travel time savings, not the frequency of limited-stop service. This can be addressed by (1) redefining a^k as a function of limited-stop frequency f, instead of a constant; and/or (2) subtracting the expected additional wait time for limited-stop service (in equivalent in-vehicle minutes) from the total travel time savings in the demand model. Although the resulting demand model will become nonlinear in f, the presented solution approach still works as it decomposes the problem into subproblems with fixed values of f.

Chapter 3

Evaluating Impacts of Integrating One-Way Vehicle-Sharing Service with Existing Public Transportation

3.1 Introduction

Vehicle-sharing service (or short-term vehicle rental service) has been growing tremendously in the past two decades and becoming an attractive alternative for urban transportation. It provides flexible, personal mobility without the full cost of vehicle ownership. As of 2010, more than 1,250,000 individuals shared 31,000 vehicles through car-sharing programs in 26 countries (Shaheen and Cohen, 2012), and as of 2011, over 236,000 bikes were installed in 135 bike-sharing programs worldwide (Shaheen et al., 2012a).

In addition to the direct benefits to individuals, vehicle-sharing service offers a number of environmental and social benefits. With a fleet of fuel-efficient, electric, or non-motorized vehicles, it reduces the transportation-related carbon footprint in a city. For car-sharing service, its cost effectiveness—achieved through ownership cost sharing—leads to car ownership reduction, which can potentially reduce traffic and parking congestion, discourage unnecessary car trips, and encourage public transportation use.

Importantly, in the case of one-way vehicle-sharing service, where users do not need to drop-off a vehicle at the pick-up station, it also complements existing public transportation systems. In particular, with a proper design of a one-way system, it can provide better access to existing public transportation and additional options for trips beyond those provided by public transit.

In spite of the many benefits of public transportation, it is economically infeasible to provide high levels of public transportation service over scattered urban areas. Additionally, the environmental benefits of public transit are only materialized when capacities are utilized to a significant extent. The average single-occupancy vehicle is more environmental friendly than the average bus carrying fewer than seven passengers (Hodges, 2010). Therefore, the potential impacts of integrating one-way vehicle-sharing service with existing public transportation can be far-reaching.

A number of studies have validated some of the benefits of vehicle sharing mentioned earlier (e.g., Rydén and Morin, 2005; DeMaio, 2009; Martin et al., 2010; Martin and Shaheen, 2011; Shaheen et al., 2012b). However, to our knowledge, no quantitative approach, beyond the use of surveys, to evaluating the impacts of the integration between one-way vehicle-sharing service and existing public transportation has been proposed. This motivates us to explore the topic further.

In this work, we present a framework for evaluating the impacts of integrated vehicle-sharing and public transportation services. The framework enables us to understand the impacts of integration in detail and answer many important questions. For example, who can benefit from a given vehicle-sharing network?; how much travel time savings can be achieved through the vehicle-sharing service?; and how do different vehicle-sharing network designs affect overall commuting patterns?

The core component of our evaluation framework is a *public transit graph* that represents a public transportation network and captures various service attributes such as wait times, travel times, and transfer times. One of the goals in developing the evaluation framework is applicability, so that our framework can be applied to transportation networks in different cities. We construct the graph using publicly available data from the General Transit Feed Specification (GTFS), which is a standard format for publishing transit service schedules, and MapQuest Open Directions Service, which is a web service that uses the open-source worldwide map provided by OpenStreetMap.

We propose three sets of metrics for measuring the impacts of vehicle-sharing service: *accessibility*, *utilization*, and *efficiency*. Similarly to centrality indices widely used in the network science literature, these new metrics are defined based on the shortest paths between transit nodes and/or vehicle-sharing stations in a network. In addition to these metrics, we develop an interactive visualization tool that helps us understand commuting patterns over a public transit network, and more importantly, identify critical transit hubs that are not well connected by the network under consideration. In order to create a compact representation of commuting patterns, we introduce the notion of a *transfer tree* that details the transfer hierarchy of trips originating from a given transit node.

Finally, as a case study, we apply the framework to assess the impact of Hubway, a bike-sharing program in the Boston metro area, on the existing public transportation system operated by the Massachusetts Bay Transportation Authority (MBTA).

This chapter is organized as follows. We first provide in Section 3.2 a review of related research in the literature. The modeling of integrated public transit and vehicle-sharing networks is described in Sections 3.3 and 3.4. In Section 3.5, we explain our evaluation metrics and visualizations of commuting patterns. In Section 3.6, we present our Boston case study. Finally, we conclude the work and discuss future work in Section 3.7.

3.2 Literature Review

In this section, we review two streams of research that are closely related to our work in this chapter.

3.2.1 Impacts of Vehicle Sharing

A number of studies have validated some of the benefits of vehicle sharing. Katzev (2003); Lane (2005); Cervero et al. (2007) report decreased vehicle miles traveled, reduced vehicle ownership, and increased public transportation usage among the members of different car-sharing programs. Based on a survey, it is estimated that car sharing in North America has removed between 90,000 to 130,000 vehicles from the road (Martin et al., 2010) and reduces the net green house gas emissions between 158,000 and 224,000 metric tons per year (Martin and Shaheen, 2011). Rydén and Morin (2005) provide a similar assessment of environmental impacts of car sharing in Europe.

For bike sharing, Borgnat et al. (2011) and Nair et al. (2013) analyze historical trip data from bike-sharing operators, Vélo'v (Lyon, France) and Vélib' (Paris, France), and allude to the use of bike-sharing service in conjunction with traditional public transit. In particular, they observe that bike stations in close proximity to transit stops and services generally have higher activity levels. The City of Paris reports that 28% of survey respondents used Vélib' as a first or last segment of their multimodal transit trips (DeMaio, 2009). Shaheen et al. (2012b) conduct a user survey across multiple bike-sharing systems in Montreal, the Twin Cities (Minneapolis and Saint Paul), Toronto, and Washington, D.C., and study, among others, the effect of bike-sharing service on traditional public transit use. The results suggest that some commuters use public transit less as their trips can be completed faster using the bike-sharing services; while others use public transportation more as the bike-sharing services provide better access to the existing transit services. The split however varies from one city to another. Importantly, 40% of the respondents indicate that they drove less as a result of bike sharing.

To our knowledge, beyond the use of surveys, no quantitative approach has been proposed to evaluating the impacts of the integration between vehicle-sharing service and existing public transportation. While a survey can potentially reveal actual user perception, the results are subjected to sampling errors, that is, the samples are not representative of the population. Long, complex surveys could also affect the accuracy of the responses and further reduce the response rate. Therefore, it is very difficult to use surveys to understand in great detail the impacts of integrated vehicle-sharing and public transportation services.

3.2.2 Transportation System Evaluation

The evaluation metrics proposed in this work are similar to *centrality indices*, which are widely used in the network science literature to study complex systems such as connection topology of social and biological networks. Centrality indices measure the importance of nodes in a network based on different criteria such as *closeness* and *betweenness* (Freeman, 1978–1979). Their applications to transportation networks were however relatively limited until the past decade.

Traditionally, centrality indices are defined for an unweighted graph. Crucitti et al. (2006) and Porta et al. (2006) extend the concept of centrality indices to geographic networks in which each arc is associated with a physical distance between the arc's end nodes, and present the Multiple Centrality Assessment framework for analyzing urban street networks.

Many published works that analyze public transit networks using centrality indices (notably, Latora and Marchiori, 2001, 2002; Sienkiewicz and Hołyst, 2005; Vragović et al., 2005; von Ferber et al., 2009; Derrible and Kennedy, 2009, 2010; Derrible, 2012) work with either unweighted graphs or simple weighted graphs in which arc lengths represent physical distances between nodes or the numbers of overlapping transit lines. They focus primarily on statistical descriptions or topological properties of the networks and ignore disutility, such as wait and transfer times, associated with trips. Additionally, most of these works are restricted to networks with a single mode of transportation.

To our knowledge, Scheurer and Porta (2006) is the first work that incorporates transit service information, such as travel times and frequencies, into the underlying network. Their transit network model is, however, different in several aspects from our approach described in Section 3.3. First, they introduce a travel arc (called *transfer-free link*) for every pair of stops along a transit line's route, as opposed to one for each pair of *adjacent* stops. With this representation, the topological length of a path (i.e., the number of arcs in the path) is equal to the number of service boardings in the corresponding trip. Because the number of arcs grows exponentially with the length of a transit line, this modeling approach can limit the capability to analyze large-scale multi-modal networks. Moreover, in their graph, boarding and transfers are not explicitly modeled with dedicated arcs, and disutility from wait time is captured by dividing the travel time associated with an arc with the service frequency. Although this approach works well for the network they consider, it may not be appropriate for networks with low-frequency transit lines because passengers might time their arrivals, and the wait times are no longer inversely proportional to service frequencies. The work is further expanded and included as part of a tool for assessing accessibility in the urban planning context (Curtis and Scheurer, 2010).

3.3 Public Transit Graph

The core component of our evaluation framework is a *public transit graph* representing a public transportation network. It captures various service attributes such as wait times, travel times, and transfer times. The graph representation allows us to apply graph-theoretic algorithms to determine the best way to travel from one location to another over the network and evaluate the network performance with respect to different metrics.

Modeling multi-modal public transit networks is quite complicated compared with road networks. Specifically, while we can simply model a road network as a graph comprising nodes and arcs representing intersections and road segments connecting the intersections (Porta et al., 2006); in a public transit graph, additional nodes and arcs are required to model multiple service lines along the same street and passenger activities like waiting and transferring. Moreover, time dependence of transit services can further complicate the resulting graph. However, for the purpose of evaluating transportation networks, we are interested in the "average" travel time over the network during a given time period. Thus, we can avoid modeling detailed service schedules and work with a more aggregated network, thereby differentiating our modeling approach from most of the published works on public transit network modeling, which focus on creating a time-expanded or time-dependent graph for trip planning purposes (Müller-Hannemann et al., 2007; Pyrga et al., 2008; Bast et al., 2010). This enables us to analyze large-scale networks.

3.3.1 Input Data

One of the goals in developing this evaluation framework is applicability—requiring minimal amounts of data and relying on publicly available data—so that our framework can be applied easily to transportation networks in different cities. In this work, we use data from the General Transit Feed Specification (GTFS), which is a standard format developed by Google for publishing transit service schedules and related transit geographic information[1]. As of 2012, more than 400 transit agencies around the world have made their service information available in the GTFS format[2].

Given a GTFS feed, we extract three elements of the public transit network: *transit nodes*, *transit lines*, and *transit services*. *Transit nodes* are access points to the public transportation service. A transit node is usually a bus stop or a rail station, but it may also represent a physical facility that contains multiple bus or rail stops. A *transit line* can be described by a sequence of stops served by the service. A *transit service* represents a group of *transit lines* that appear as a single service to commuters. For instance, the MBTA's bus 1 service has two lines serving two different sequences of stops—inbound and outbound. Some transit services may operate different sets of transit lines at different times of the day.

The sets of transit nodes and transit services, denoted by $\mathcal{N}$ and $\mathcal{M}$, are readily available from files `stops.txt` and `routes.txt` in the GTFS feed. For each service $m \in \mathcal{M}$, we extract trips associated with the service from file `trips.txt` and group them into transit lines based on their stop sequences (detailed in file

[1] http://developers.google.com/transit/gtfs/reference

[2] http://www.gtfs-data-exchange.com/

`stop_events.txt`). The result is the set of transit lines, $\mathcal{L}$ where each line $l \in \mathcal{L}$ belongs to transit service $m(l) \in \mathcal{M}$. Additionally, we use the time information associated with trips to compile service schedule information for each transit line, including departure and arrival times at every stop in the sequence, average frequency/headway, and travel times between stops.

3.3.2 Graph Construction

Figure 3-1 illustrates the structure of our public transit graph. Given public transit service information extracted from a GTFS feed, we construct a directed graph representing the transit system as follows.

Nodes

There are two sets of nodes in the public transit graph—one associated with transit nodes and the other associated with transit lines. For each transit node $n \in \mathcal{N}$, we introduce *access* and *egress* nodes, denoted by $acc(n)$ and $egr(n)$, where passengers begin and end their trips.

For each transit line $l \in \mathcal{L}$ described by a sequence of $|l|$ stops $(s^l_1, s^l_2, \ldots, s^l_{|l|})$, we add a *depart* node, $dep(l, s^l_i)$ for each $i \in \{1, 2, \ldots, |l| - 1\}$ (i.e., every stop except the *last* stop in the sequence) and add an *arrive* node, $arr(l, s^l_i)$ for each $i \in \{2, 3, \ldots, |l|\}$ (i.e., every stop except the *first* stop in the sequence).

Arcs

There are six types of arcs representing activities and connectivities over the public transit network.

1. *Walk Arcs* (Figure 3-2a). For short trips, passengers may walk directly from one transit node to another. We therefore connect each access node to egress nodes of nearby transit nodes. Let $\mathcal{N}_n$ denote the set of transit nodes within a maximum walking distance from a given transit node n. Formally, the set of walk arcs is given

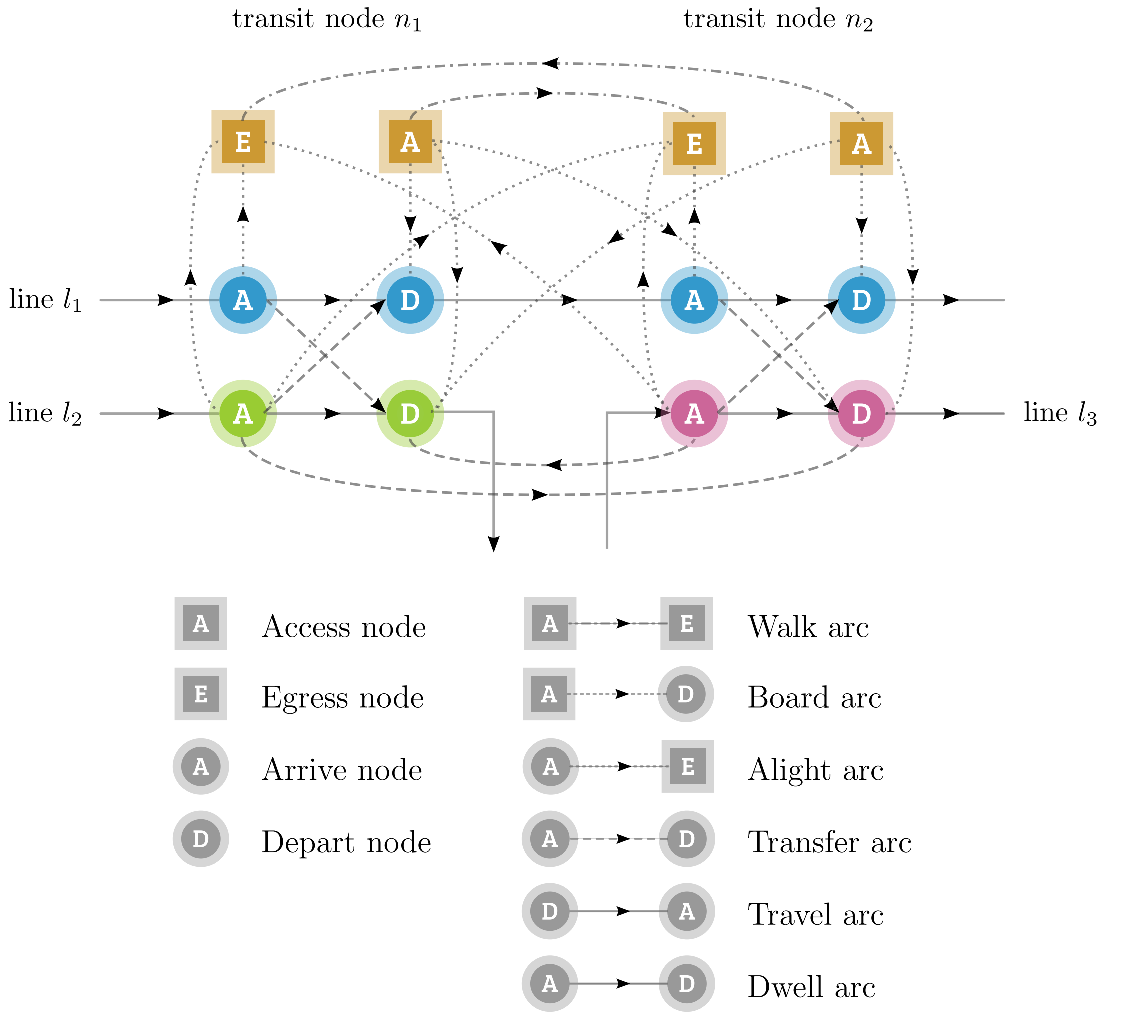
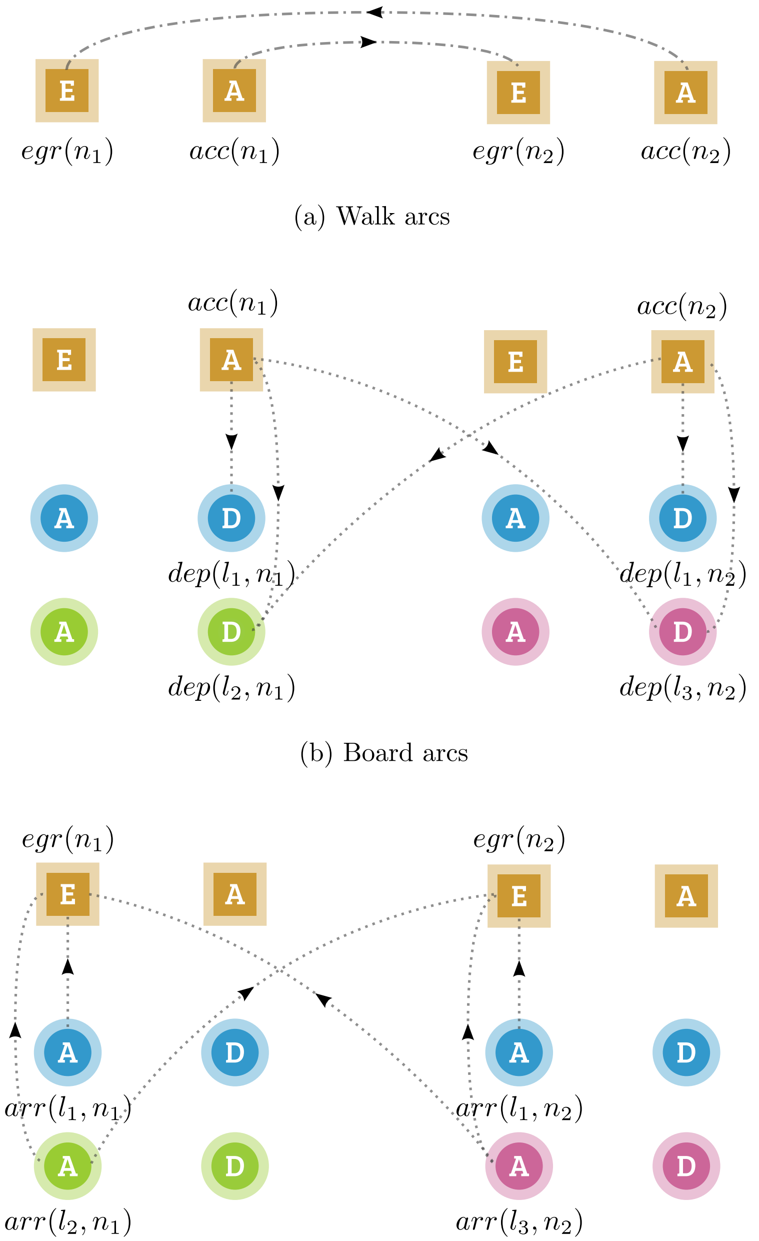


Figure 3-1: An example of public transit subgraph induced by two nearby stops served by three transit lines.

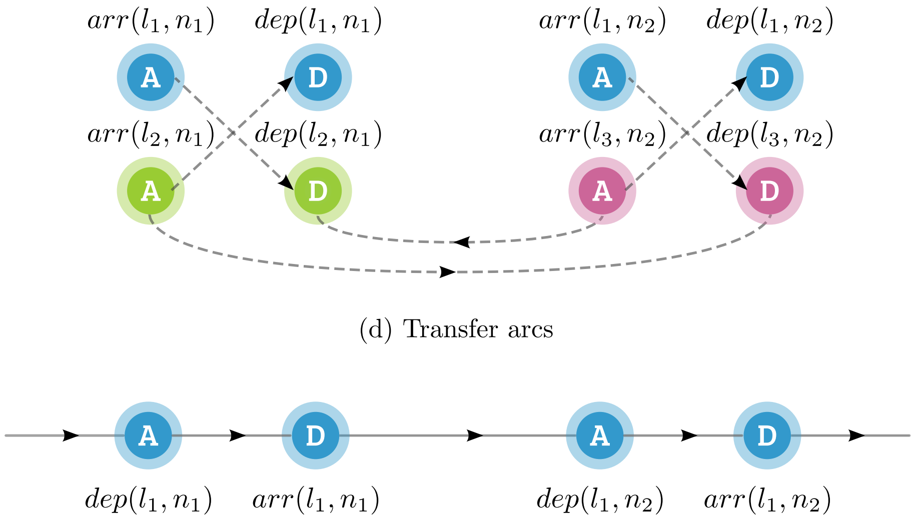


(a) Walk arcs

(b) Board arcs

(c) Alight arcs

Figure 3-2: Arcs in public transit graph



(d) Transfer arcs

(e) Travel and dwell arcs (for line l_1)

Figure 3-2: Arcs in public transit graph

by

$$\{(acc(n), egr(n')) \mid n \in \mathcal{N}, n' \in \mathcal{N}_n\}.$$

2. *Board Arcs* (Figure 3-2b). Connecting access to depart nodes, a board arc represents a passenger waiting and then boarding a transit line. It may connect access and depart nodes of different transit nodes, that is a passenger can also walk to a nearby transit node, wait, and then board a transit line that does not serve his/her access point. Let $\mathcal{L}_n$ denote the set of transit lines serving a given transit node n. Mathematically, the set of board arcs is given by

$$\begin{aligned} &\{(acc(n), dep(l, n)) \mid n \in \mathcal{N}, l \in \mathcal{L}_n, n \neq s^l_{|l|}\} \\ \cup\ &\{(acc(n), dep(l, n')) \mid n \in \mathcal{N}, n' \in \mathcal{N}_n, l \in \mathcal{L}_{n'} \setminus \mathcal{L}_n, n' \neq s^l_{|l|}\}. \end{aligned}$$

Note that passengers cannot board transit line l at its last stop $s^l_{|l|}$. Condition $l \in \mathcal{L}_{n'} \setminus \mathcal{L}_n$ ensures that a passenger walks to board a transit line at a nearby transit node only if the transit line does not serve his/her access point.

3. *Alight Arcs* (Figure 3-2c). In parallel with board arcs, we have alight arcs

connecting arrive to egress nodes. The set of alight arcs is given by

$$\begin{aligned}&\{(arr(l,n), egr(n)) \mid n \in \mathcal{N}, l \in \mathcal{L}_n, n \neq s_1^l\}\\ \cup\ &\{(arr(l,n'), egr(n)) \mid n \in \mathcal{N}, n' \in \mathcal{N}_n, l \in \mathcal{L}_{n'} \setminus \mathcal{L}_n, n' \neq s_1^l\}.\end{aligned}$$

Note that passengers cannot alight from transit line l at its first stop s_1^l.

4. *Transfer Arcs* (Figure 3-2d). Passengers can transfer from one transit line to another of a different transit service, probably of different modes, through a transfer arc connecting an arrive node of the inbound transit line to a depart node of the outbound transit line. A transfer may occur within a transit node or between two nearby transit nodes. Formally, the set of transfer arcs is given by

$$\begin{aligned}&\{(arr(l,n), dep(l',n) \mid n \in \mathcal{N}, l, l' \in \mathcal{L}_n, m(l) \neq m(l'), n \neq s_1^l, n \neq s_{|l'|}^{l'}\}\\ \cup\ &\{(arr(l,n), dep(l',n') \mid n \in \mathcal{N}, n' \in \mathcal{N}_n, l \in \mathcal{L}_n, l' \in \mathcal{L}_{n'} \setminus \mathcal{L}_n,\\ &\qquad\qquad m(l) \neq m(l'), n \neq s_1^l, n' \neq s_{|l'|}^{l'}\}.\end{aligned}$$

Condition $m(l) \neq m(l')$ ensures that transit lines l and l' belong to different transit services $m(l)$ and $m(l')$.

5. *Travel Arcs* (Figure 3-2e). A travel arc represents a passenger's in-vehicle movement from one transit stop to the next one served by the transit line. It connects a depart node of a transit line at a stop to the arrive node at the next stop. The set of travel arcs is given by

$$\{(dep(l, s_i^l), arr(l, s_{i+1}^l)) \mid l \in \mathcal{L}, i \in \{1, 2, \ldots, |l| - 1\}\}.$$

6. *Dwell Arcs* (Figure 3-2e). Between travel arcs, there are dwell arcs connecting arrive and depart nodes at the same stop. A dwell arc captures a period when passengers get on and off the vehicle or wait on the vehicle. The set of dwell arcs is given by

$$\{(arr(l, s_i^l), dep(l, s_i^l)) \mid l \in \mathcal{L}, i \in \{2, 3, \ldots, |l| - 1\}\}.$$

Given the public transit graph, a route from transit nodes n to n' can be represented as a direct path from the access node $acc(n)$ to the egress node $egr(n')$.

Notice that another possible way to model a passenger's walking and boarding a transit line at nearby transit nodes is to have walk arcs connect to access nodes, so that passengers can first walk from one access node to another and then board a transit line, as presented in Spiess and Florian (1989). Essentially, this enables us to reduce the number of arcs by replacing all board arcs across transit nodes—one for each transit line—with walk arcs to the access nodes—one for each nearby transit node. This alternative modeling approach, however, allows a path visiting several access nodes constituting a walking path that may exceed the given maximum walking distance. We therefore do not adopt this approach to avoid solving the constrained shortest path problem when we evaluate the network. A similar reasoning applies to our modeling approach for alight and transfer arcs across transit nodes.

Additionally, it is important to note that our graph has one transfer arc for each pair of transit lines, which allows us to capture connection-specific information, such as mode switch or a minimum transfer time requirement. As shown in Spiess and Florian (1989), another way to model transfers, potentially with fewer arcs, is to replace transfer arcs with alight arcs connecting arrive to access nodes, in addition to egress nodes. Consequently, a transfer can be represented by a path from an arrive node to an access node through an alight arc, and then to a depart node through a board arc. This alternative approach, in addition to not capturing connection-specific information, may lead to less accurate calculation of transfer times as it is implicitly assumed that the time until the next outbound service's departure is independent of the inbound service.

3.3.3 Costs

There are many factors that may affect a passenger's route choice (modes, transit lines, and transfer points). In order to compare different public transit routes, we associate each arc in the public transit graph with a vector of various types of costs and define a cost function, or a utility function (Ben-Akiva and Lerman, 1985), specifying the weights for different cost components. A passenger is then assumed to take the route whose associated path yields the least total cost.

In this work, we consider three types of costs: average travel time, transfer penalty, and schedule delay.

Average Travel Time

Travel time can be categorized into in-vehicle time, walk time, and wait time.

1. *In-vehicle travel times.* In-vehicle travel times are associated with arcs representing vehicle movement, that is, dwell and travel arcs. The average in-vehicle time for each arc can be obtained directly from the transit line's schedule.

2. *Walk times.* All arcs that include walking have an associated positive walk time. In addition to walk arcs, these include board, alight, and transfer arcs across transit nodes. For simplicity, we estimate walk time using a great-circle distance[3] and assuming a walking speed of 5 kilometers/hour (about 3 miles/hour).

3. *Wait times.* There are wait times associated with board and transfer arcs. For the purpose of computing average wait times, we assume that the transit operator strives to maintain regular headways for high-frequency, operating every 15 minutes or less, lines and adheres to published schedules for low-frequency lines. More specifically, the arrival process of a high-frequency line is a random process that is independent of arrival processes of other transit lines, with a deterministic interarrival time equal to the average headway of the service; while the arrival process of a low-frequency line is deterministic according to the published schedule. Given these assumptions, we presume that passengers whose transit trips include low-frequency service, not necessarily in the first segment of the trip, will time their trips to minimize the wait times.

The calculations of wait times for board and transfer arcs are different as wait times for the latter may also depend on the inbound transit line, as opposed to just

[3] the shortest distance between two points on the surface of a sphere

the outbound line. For each board arc, we assume wait time is half the headway if the transit line runs at a high frequency. Otherwise, we assume that passengers will time their arrivals, and their wait times are fixed at 7.5 minutes. The reason for this rather high fixed wait time for low-frequency lines is that passengers do not want to miss their connections and thus, arrive early to avoid at least another 15 minutes of waiting. (We acknowledge that in practice, this buffer time should be set based on the reliability of a transit line.) In summary, a wait time associated with a board arc to transit line l is given by $\min(h_l/2, 7.5)$ where h_l is the headway of transit line l in minutes.

Now consider an average wait time of a transfer between transit lines l and l' with headways of h_l and $h_{l'}$ minutes, respectively. If the outbound transit line l' is a high-frequency service, the average wait time is given by $h_{l'}/2$ because of our independence assumption for high-frequency lines. For a transfer from a high-frequency line to a low-frequency line, we presume that passengers will time the inbound trip on a high-frequency line to avoid arriving at the transfer point too early. As a result, the wait time is uniformly distributed between 0 and h_l, and the average wait time is given by $h_l/2$, half the headway of the inbound service. Intuitively, the higher frequency of the inbound service, the shorter the wait time at the transfer point.

Lastly, for a transfer between two low-frequency lines, despite the inflexibility, we presume that passengers will still try to minimize the wait time. In particular, we calculate the average wait time considering only *efficient connections*, defined as those with minimum wait times locally. Formally, let $\{a_1^l, a_2^l, \ldots, a_{f_l}^l\}$ be an ordered set of arrival times of transit line l at an inbound stop and $\{d_1^{l'}, d_2^{l'}, \ldots, d_{f_{l'}}^{l'}\}$ be an ordered set of departure times of transit line l' at an outbound stop, which may or may not be the same as the inbound stop. A connection $(a_i^l, d_j^{l'})$ is a feasible connection if the time difference $d_j^{l'} - a_i^l$ is larger than a minimum transfer time requirement, δ. Minimum transfer times are equal to walk times for transfers across transit nodes. For a transfer within a transit node, a minimum transfer time represents the amount of time that passengers need to walk between platforms and can be obtained from file `transfers.txt` in the GTFS feed. Figure 3-3 illustrates feasible and efficient

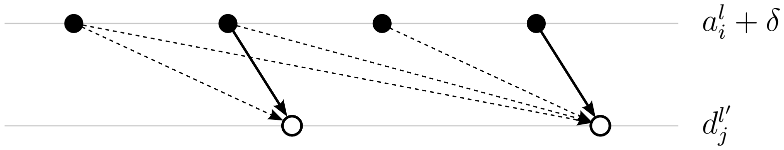


Figure 3-3: Connections between two low-frequency transit lines. Dashed arrows represent *feasible connections*, and solid arrows represent *efficient connections*.

connections between two low-frequency transit lines. Mathematically, an *efficient connection* $(a_i^l, d_j^{l'})$ is a feasible connection that satisfies the following condition:

$$\begin{aligned} \nexists i' \in \{i+1, i+2, \ldots, f_l\}, \quad d_j^{l'} - a_{i'}^l \geq \delta \text{ and} \\ \nexists j' \in \{1, 2, \ldots, j-1\}, \quad d_{j'}^{l'} - a_i^l \geq \delta. \end{aligned}$$

Transfer

Because of inconvenience associated with transfers, passengers generally prefer transit routes with fewer or no transfers, unless the alternatives with more transfers can provide significant travel time savings. In this work, we penalize transfers by assigning a fixed cost to each transfer arc. Note that this transfer penalty is imposed in addition to the travel time associated with the transfer, such as wait time for the outbound service and/or walk time to the another transit node.

Schedule Delay

Although passengers can time their trips involving low-frequency transit lines to minimize the total travel time, low-frequency services are not preferable due to the lack of flexibility and the hassle of the need to time trips. To penalize for this inconvenience, we assign a fixed cost to board and transfer arcs associated with boarding low-frequency transit lines.

Lastly, transit fare is not considered in this work, as commuters may purchase a discounted pass for unlimited rides and hence make a route choice without taking fare into account. It however can be included in the model by assigning fare to board and transfer arcs accordingly. Moreover, because our transit graph can capture connection-specific information through transfer arcs, we may adopt a more sophisticated model of transfer penalties, including those recently presented in Guo and Wilson (2011) and Raveau et al. (2011), that distinguish transfers between different modes, at different stations, with different facilities (such as staircase, escalators), etc.

3.4 Vehicle Sharing Network

In order to consider integrated public transit and vehicle-sharing services, we augment our transit graph with nodes and arcs corresponding to vehicle-sharing options.

Let $\tilde{\mathcal{N}}$ be the set of vehicle-sharing stations. For each station $\tilde{n} \in \tilde{\mathcal{N}}$, we introduce access and egress nodes where passengers begin and end their trips. To be consistent with our public transit representation, a vehicle-sharing service is modeled as a single transit line, denoted by $\tilde{l}$, and we add a depart node $dep(\tilde{l}, \tilde{n})$ and an arrive node $arr(\tilde{l}, \tilde{n})$ for each station $\tilde{n} \in \tilde{\mathcal{N}}$.

Walk, board, alight and transfer arcs can then be added in a manner similar to that for public transit. For tractability, a trip from one vehicle-sharing station to another is represented by a single travel arc connecting the origin's depart node to the destination's arrive node. This representation allows us to avoid modeling the underlying road network, thereby increasing significantly the sizes of the networks we are able to evaluate, without sacrificing our network evaluation capabilities. Unlike transit lines, which operate on fixed routes, there can be multiple travel arcs adjacent to arrive and depart nodes of the vehicle-sharing service, and there are no dwell arcs.

Because of the on-demand nature of vehicle-sharing services, there are no wait times associated with board and transfer arcs. Nevertheless, we assign board, alight, and transfer arcs each a fixed amount of travel time corresponding to renting and returning a shared vehicle.

Travel time between stations can be estimated using historical trip data specifying origin, destination, and duration of each trip made by users. However, when this evaluation framework is applied to assess a new vehicle-sharing network, such data might not be available. In this case, we can utilize existing trip planners to obtain the best route—presumably taken by most commuters—between each pair of stations. In this work, we use MapQuest Open Directions Service[4], a publicly available web service that uses the open-source worldwide map provided by OpenStreetMap[5].

3.5 Network Evaluation

In this section, we describe various metrics and a visualization technique that we use to analyze multi-modal transportation networks and evaluate the impacts of integrating vehicle-sharing service with existing public transportation.

3.5.1 Metrics

In this work, we propose three sets of metrics that measure the impacts of vehicle-sharing service: *accessibility*, *utilization*, and *efficiency*. Similarly to centrality indices, these new metrics are defined based on the shortest (least cost) paths—minimizing overall disutility—between transit nodes and/or vehicle-sharing stations in a network. Centrality indices are not directly applicable here because they are defined for an individual node in a network, while we are interested in metrics defined for a transit node or a vehicle-sharing station, which are associated with multiple nodes in our public transit graph.

Accessibility

One immediate question one might have when given a transportation network is how easy it is to access other transit nodes from a given node. We measure accessibility of a transit node by calculating the average travel time and the average number of transfers of the shortest paths to every other transit nodes.

[4] http://open.mapquestapi.com/directions/

[5] http://wiki.openstreetmap.org/

Let $tt(i,j)$ and $tr(i,j)$ denote the total travel time and the number of transfers of the shortest path from nodes i to j. The average travel time from a transit node $n \in \mathcal{N}$ to other transit nodes is given by

$$A_{tt}(n) = \frac{1}{|\mathcal{N}|-1} \sum_{n' \in \mathcal{N} \setminus \{n\}} tt(acc(n), egr(n')), \tag{3.1}$$

and the average number of transfers from a transit node $n \in \mathcal{N}$ to other transit nodes is given by

$$A_{tr}(n) = \frac{1}{|\mathcal{N}|-1} \sum_{n' \in \mathcal{N} \setminus \{n\}} tr(acc(n), egr(n')). \tag{3.2}$$

The larger the values of $A_{tt}(n)$ and $A_{tr}(n)$, the poorer the accessibility to other transit nodes from node n. Note that the metrics consider only origin-destination (O-D) pairs between public transit nodes, not vehicle-sharing stations. This is to ensure that when we evaluate different designs of vehicle-sharing networks, the metrics are directly comparable, as the same set of trips is considered. Additionally, the metrics are only well defined for a connected graph where $tt(acc(n), egr(n'))$ and $tr(acc(n), egr(n'))$ are finite. Our accessibility metrics are similar to the *closeness centrality* indices of Freeman (1978–1979).

At the network level, the average travel time and the average number of transfers of all O-D pairs between public transit nodes are given by

$$A_{tt}(\mathcal{N}) = \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} tt(acc(n), egr(n')), \text{ and} \tag{3.3}$$

$$A_{tr}(\mathcal{N}) = \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} tr(acc(n), egr(n')). \tag{3.4}$$

Utilization

For a given design of a vehicle-sharing network, it is important to know the potential service usage. Specifically, we want to know the proportion of O-D pairs whose shortest paths utilize vehicle-sharing service. This can be achieved by calculating the proportion of O-D pairs whose shortest paths contain a segment using the vehicle-sharing service.

Formally, for each vehicle-sharing link from vehicle-sharing node $\tilde{n}$ to $\tilde{n}' \in \tilde{\mathcal{N}}$, its potential utilization is given by

$$U(\tilde{n}, \tilde{n}') = \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} \mathbb{1}_{(dep(\tilde{l},\tilde{n}), arr(\tilde{l},\tilde{n}')) \in r_{n,n'}}, \tag{3.5}$$

where $r_{n,n'}$ denotes the shortest path between nodes n and n'. Each term in the summation equals one if the corresponding shortest path contains the travel arc between the vehicle-sharing stations $(dep(\tilde{l}, \tilde{n}), arr(\tilde{l}, \tilde{n}'))$, and zero otherwise. Again, we only count the O-D pairs from the set of public transit nodes $\mathcal{N}$.

At the station level, the potential utilization of each vehicle-sharing station $\tilde{n} \in \tilde{\mathcal{N}}$ is given by

$$\begin{aligned} U(\tilde{n}) &= \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} \mathbb{1}_{dep(\tilde{l},\tilde{n}) \in r_{n,n'} \vee arr(\tilde{l},\tilde{n}) \in r_{n,n'}} \\ &= \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} \left[\mathbb{1}_{dep(\tilde{l},\tilde{n}) \in r_{n,n'}} + \mathbb{1}_{arr(\tilde{l},\tilde{n}) \in r_{n,n'}} \right] \\ &= \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} \left[\sum_{\tilde{n}' \in \tilde{\mathcal{N}} \setminus \{\tilde{n}\}} \mathbb{1}_{(dep(\tilde{l},\tilde{n}), arr(\tilde{l},\tilde{n}')) \in r_{n,n'}} \right. \\ &\qquad \left. + \sum_{\tilde{n}' \in \tilde{\mathcal{N}} \setminus \{\tilde{n}\}} \mathbb{1}_{(dep(\tilde{l},\tilde{n}'), arr(\tilde{l},\tilde{n})) \in r_{n,n'}} \right] \\ &= \sum_{\tilde{n}' \in \tilde{\mathcal{N}} \setminus \{\tilde{n}\}} U(\tilde{n}, \tilde{n}') + U(\tilde{n}', \tilde{n}). \end{aligned} \tag{3.6}$$

The second equality follows from the fact that the shortest path $r_{n,n'}$ does not contain a loop and may include either the depart node $dep(\tilde{l}, \tilde{n})$ or the arrive node $arr(\tilde{l}, \tilde{n})$, not both. This metric is similar to the *betweenness centrality* indices (Freeman, 1977).

Lastly, the potential utilization of a given set of stations $\tilde{\mathcal{N}}$, is given by

$$U(\tilde{\mathcal{N}}) = \frac{1}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{\{n,n' \in \mathcal{N} | n \neq n'\}} \mathbb{1}_{r_{n,n'} \cap \{(dep(\tilde{l},\tilde{n}), arr(\tilde{l},\tilde{n}')) | \tilde{n}, \tilde{n}' \in \tilde{\mathcal{N}}\} \neq \emptyset}. \tag{3.7}$$

Each term in the summation is one if the corresponding shortest path contains at least one travel arc using the vehicle-sharing service. Consequently, each O-D pair can be counted at most once even if its shortest path may consist of multiple segments using the vehicle-sharing service.

Efficiency

Passengers will find a link, provided by a vehicle-sharing service, from one station to another attractive if it is more efficient than existing public transit, that is providing travel time savings or reducing the number of transfers required. In order to measure travel time savings and transfer reduction, we need to compare the shortest paths between the vehicle-sharing stations obtained from networks with and without the vehicle-sharing service.

Let G_0 be a public transit graph without a vehicle-sharing service. Specifically, arrive nodes, depart nodes, and travel arcs associated with the vehicle-sharing service are removed, but egress and access nodes corresponding to vehicle-sharing stations are retained. This allows us to consider trips starting and ending at vehicle-sharing stations without using the vehicle-sharing service. Let $tt_0(i,j)$ and $tr_0(i,j)$ denote the total travel time and the number of transfers of the shortest path from nodes i to j over graph G_0.

Travel time savings provided by a link between vehicle-sharing stations $\tilde{n}$ and $\tilde{n}' \in \tilde{\mathcal{N}}$ is given by

$$E_{tt}(\tilde{n},\tilde{n}') = \text{Max}(0, tt_0(acc(\tilde{n}), egr(\tilde{n}')) - t_{\tilde{n},\tilde{n}'}), \tag{3.8}$$

where $t_{\tilde{n},\tilde{n}'}$ is the travel time from station $\tilde{n}$ to $\tilde{n}'$ using vehicle-sharing service. We define travel time savings to be nonnegative because passengers can use public transit if the link between vehicle-sharing stations does not provide a shorter travel time.

Because the vehicle-sharing service provides a direct link between stations $\tilde{n}$ and

$\tilde{n}' \in \tilde{\mathcal{N}}$, the reduction in the number of transfers is simply given by

$$E_{tr}(\tilde{n}, \tilde{n}') = tr_0(acc(\tilde{n}), egr(\tilde{n}')). \tag{3.9}$$

For efficiency at any station $\tilde{n}$ and at the network level, we calculate the average efficiency over the links involving station $\tilde{n}$ and over every link between vehicle-sharing stations, respectively.

3.5.2 Visualization

In addition to measuring the impacts of vehicle-sharing service using the accessibility, utilization, and efficiency metrics, it is necessary to understand commuting patterns over an existing public transit network.

We first need to identify the shortest paths between transit nodes in a network. Algorithms like Dijkstra's algorithm (Dijkstra, 1959) or its variants (see Ahuja et al., 1993) can be used to obtain the shortest paths from an origin to the other nodes in the network. These algorithms output a so-called *shortest path tree*, a directed tree detailing the unique optimal paths from the root (origin) to the other nodes (see Figure 3-4a). Essentially, a shortest path tree reveals commuting patterns over the transit network, key transit lines that connect the origin to destinations, and major stops where commuters make transfers.

It is however difficult to create an intelligible visualization of a shortest path tree for our public transit graph because there are multiple nodes associated with each transit node, and the number of transit nodes in a city can be very large. One possible solution is to collapse all nodes corresponding to the same transit node into a single node. Nevertheless, we are likely to lose the tree structure due to overlapping transit lines (see Figure 3-4b). In particular, there may exist multiple arcs connecting some pairs of nodes in the resulting graph, and therefore, we can no longer accurately identify from the graph the shortest paths from the origin to the other transit nodes.

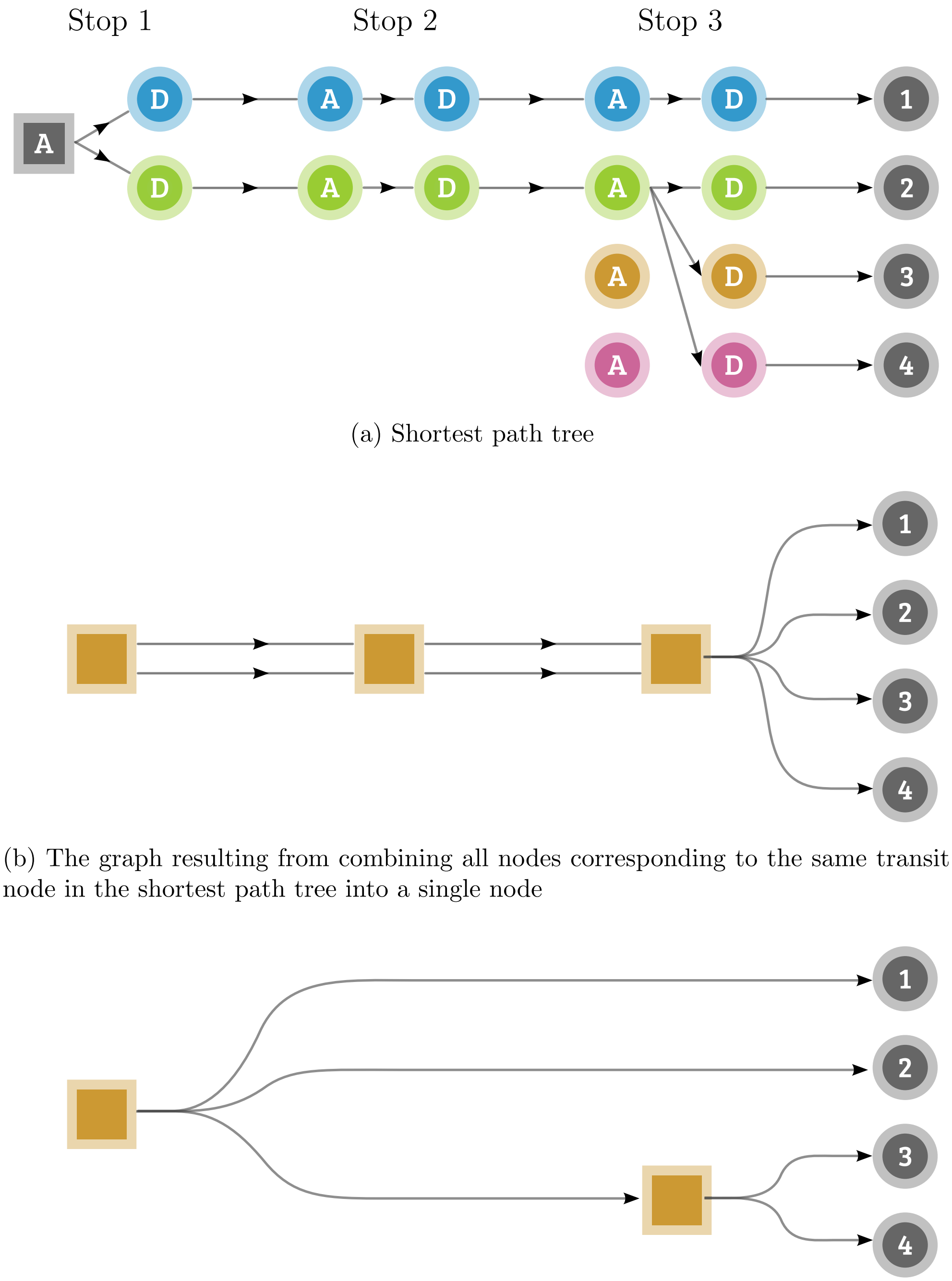
To overcome this difficulty, we propose the notion of *transfer tree*. Instead of including every transit node, a transfer tree includes only the origin and the transfer points—transit nodes at which commuters board subsequent transit services in their shortest paths. Each arc in a transfer tree represents a transfer-free movement from one transit point (or the origin) to another transit point. Because the shortest path from the origin to each transfer point is the same for any destinations reached via the transfer point, there is exactly one incoming arc to each transfer point, and therefore, the resulting transfer tree inherits the tree structure (see Figure 3-4c).

To highlight the importance of each transfer point, we can visualize transfer trees using the Sunburst visualization (Stasko et al., 2000), as illustrated in Figure 3-5a. Located at the center of the plot is the root of a transfer tree (an origin location). The circular slices in the innermost ring represent the access points, where commuters board their first transit services. Circular slices representing subsequent transfer points are placed outward from the center. For instance, the second innermost ring represents first transfer points in the optimal paths. The underlying tree structure is shown in Figure 3-5b. The angle swept by each slice (hence, the area) is proportional to the number of destinations whose shortest paths visit the corresponding transit node. And the color of each slice indicates the travel time from the origin to the corresponding transit node.

Depicting the transfer hierarchy of trips originating from a given transit node, this compact representation allows us to compare commuting patterns of different origins, and more importantly, identify critical transit hubs that are not well connected by the network under consideration—big slices that are located further away from the center of the plot. More examples and demonstration of its use will be given in the case study.

3.6 Case Study: Boston Network

To demonstrate use of our evaluation framework, we assess the impact of Hubway, Boston's bike-sharing program, on the existing public transportation system operated by the Massachusetts Bay Transportation Authority (MBTA).

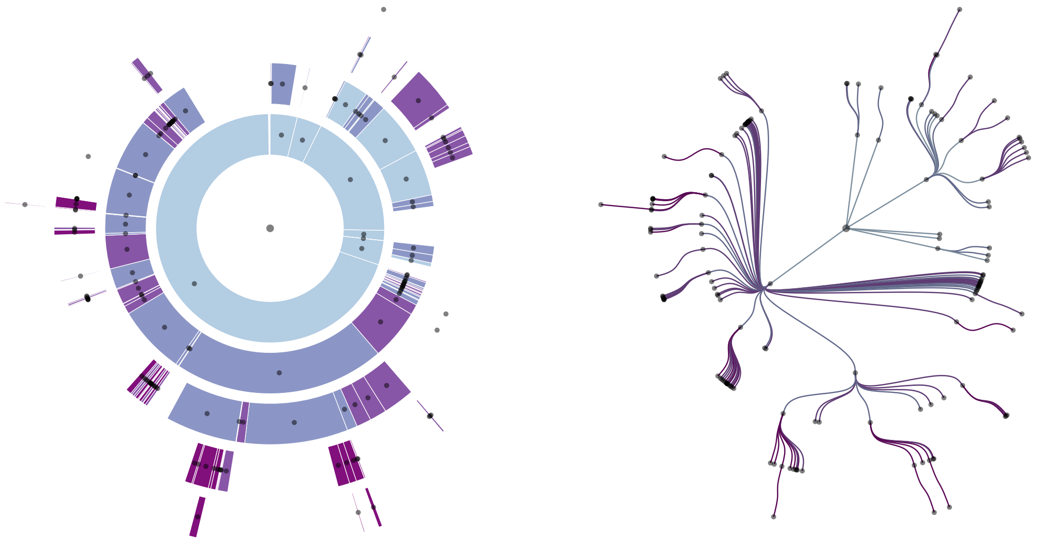


(a) Shortest path tree

(b) The graph resulting from combining all nodes corresponding to the same transit node in the shortest path tree into a single node

(c) The corresponding transfer tree

Figure 3-4: In this example illustrating a shortest path tree and a transfer tree, there are three stops and four transit lines with four different destinations. From the shortest path tree, passengers traveling to destinations 3 and 4 need to make a transfer at Stop 3.



(a) (b)

Figure 3-5: Transfer tree visualization

3.6.1 Network Details and Input Data

The MBTA provides public transportation service to 175 cities and towns in the Boston Metro area (MBTA, 2010). Spanning over 1,500 route miles, the MBTA services include subway (the T), bus, commuter rail, and boat. The key service map is shown in Figure 3-6. As discussed in section 3.3.1, we obtain the service information from the GTFS feed provided by the MBTA. In this work, we focus on the service network over the weekday morning peak (6–9 am), which contains 8,120 transit nodes, 234 transit services, and 683 transit lines (around 15% of which are high-frequency lines, operating every 15 minutes or less).

Hubway[6] was introduced in July, 2011 with an initial fleet of 600 bicycles and 60 stations located around Boston. To access the fleet of bicycles, users may register for an annual membership or purchase a 3-day or 1-day pass. This covers unlimited short trips, and any trips longer than 30 minutes incur additional fees. Like most bike-sharing programs, a user can rent a bike from any station and return it to another. This one-way rental model allows commuters to use shared bikes in conjunc-

[6] http://thehubway.com/

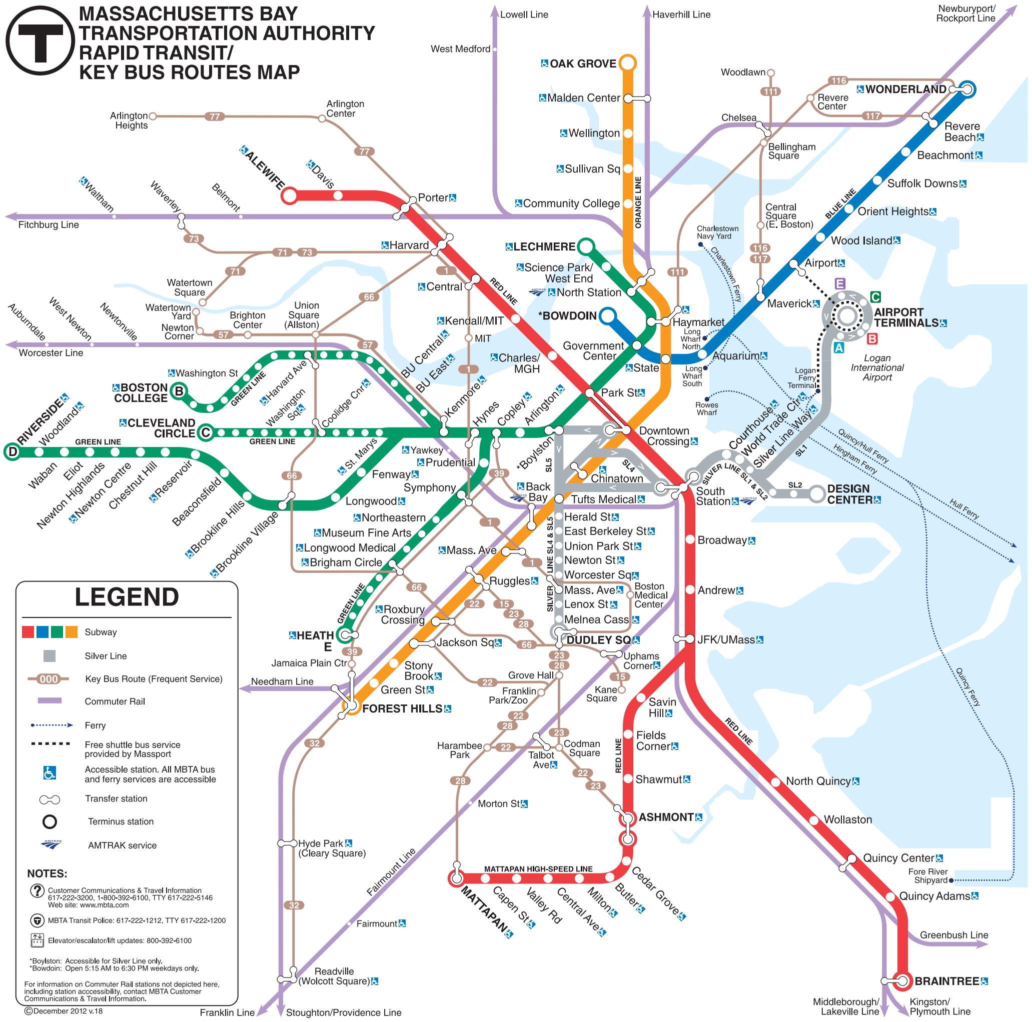
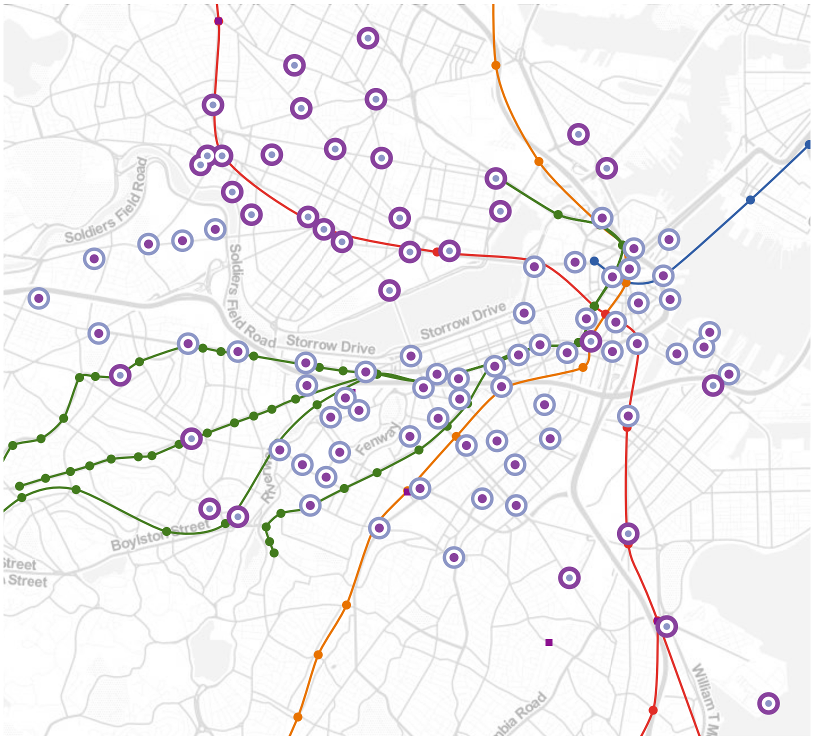


Figure 3-6: An MBTA rapid transit/key bus routes map

tion with traditional transit services. In Summer 2012, Hubway partnered with the municipalities of Brookline, Cambridge, and Somerville and expanded their network to over 100 stations.

The Hubway's network information can be obtained from their live feed[7], listing stations together with other station-specific information such as location, number of available bikes/docks, and installed date. In this work, we consider two Hubway networks: one comprising 61 stations operated at the end of 2011 and the other comprising 95 stations operated at the end of September, 2012 (see Figure 3-7).

[7] http://www.thehubway.com/data/stations/bikeStations.xml



Station operated since ◉ 2011 ◉ 2012

Figure 3-7: A Hubway station map

In addition to the live feed, as part of the Hubway Data Visualization Challenge[8], Hubway released trip historical data, including dates, durations, origin and destination stations of every trip made from the time of Hubway's launch until the end of September, 2012. With this dataset, we validate some of our findings and understand the limitations of our evaluation framework.

We obtain bike times using MapQuest Open Directions service (MQOD), as discussed in Section 3.4, and calibrate them using the trip historical data. In particular, for each O-D pair that has at least 20 trips made by registered users, we estimate the bike time using the median of durations of trips made by registered users. We focus on registered users as they are likely to use the service for commuting purposes, and their trip durations should be less variable. From the dataset, we can calculate bike time estimates for about 30% of all station pairs. We then compare the estimates with the bike times obtained from MQOD. In Figure 3-8, we can see that the bike

[8] http://hubwaydatachallenge.org/

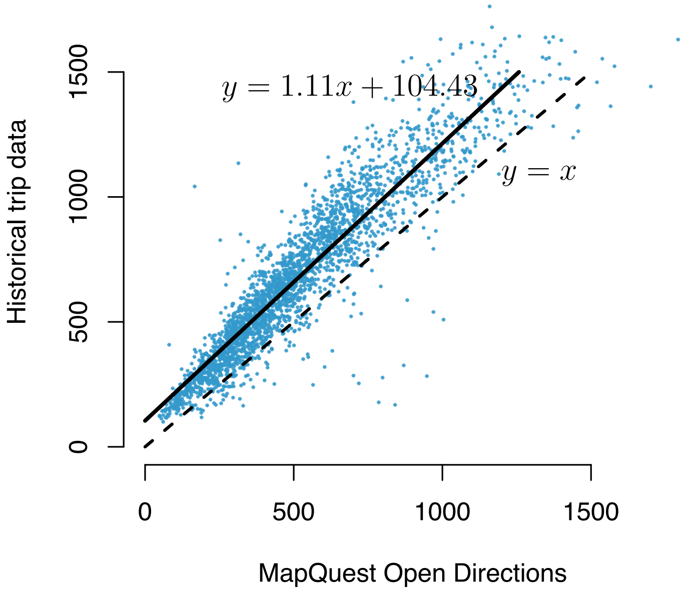


Figure 3-8: Bike times between stations (in seconds)

	MBTA	with Hubway (2011)	with Hubway (2012)
Nodes	56,710	56,764	56,900
Arcs	2,630,987	2,649,081	2,672,560

Table 3.1: Graph sizes

times from MQOD consistently underestimate the bike times from the historical data. To account for this, we fit a simple linear model and use it to adjust the bike times of every O-D pair from MQOD.

Given the network information, we build the public transit graphs as outlined in Sections 3.3 and 3.4. We assume, in this work, the maximum walking distance is 800 meters (about half a mile), and the maximum bike trip duration is 30 minutes. The former directly affects the number of arcs involving walking across transit nodes in the graphs, while the latter affects the number of travel arcs between bike stations. The sizes of the resulting graphs are summarized in Table 3.1.

Note that in the MBTA-only graph, we also include access and egress nodes associated with every bike station, so that we can use the graph to compute efficiency metrics, as discussed in Section 3.5.1.

3.6.2 Evaluation

In order to compute the evaluation metrics presented in Section 3.5, we first need to obtain the shortest paths between transit nodes. We implement, in Python, Dijkstra's algorithm using the heap (priority queue) data structure to speed up computational times (see Ahuja et al., 1993). The algorithm is then used repeatedly to solve one-to-all shortest path problems starting at each transit node. For each instance with over 8,000 transit nodes, obtaining the shortest paths for every O-D pair takes around one hour on a Mac OS X machine with an Intel Core i7 2.7 GHz processor and 8 GB of RAM.

The cost function we use in this work is relatively simple. In particular, different types of travel times (in-vehicle, walk, and wait) are weighted equally, and inconvenience due to transfers or schedule delay is penalized—each transfer and low-frequency service boarding is equivalent to 10 and 5 minutes of travel time, respectively. For interested readers, Central Transportation Planning Staff (1997), Guo and Wilson (2004), and Guo and Wilson (2007) have extensively studied the cost of transfers for the Boston metro area and estimated route choice models using data from an on-board survey conducted in 1994. None of these works, however, include cycling as an alternative.

Network Level

Table 3.2 summarizes the impact of Hubway at the network level. The Hubway service in 2011 reduces the average travel time and the average number of transfers for O-D pairs between transit nodes (about 66 million O-D pairs) by 0.57% and 0.43%, respectively. This rather small improvement in network accessibility is because there are a large number of O-D pairs whose origins and destinations are outside the Hubway service area. In fact, while each link between bike-stations, on average, provides travel time savings of almost 9 minutes and decreases the number of transfers by 0.208, only 8.43% of O-D pairs benefit from the Hubway service (that is, utilize the service in their shortest paths).

	MBTA	with Hubway	
		2011	2012
Accessibility			
Avg. travel time (secs)	4,267	4,243 (-0.57%)	4,207 (-1.40%)
Avg. number of transfers	1.764	1.756 (-0.43%)	1.725 (-2.20%)
Utilization			
% of O-D pairs that benefit from the Hubway service	–	8.43	17.74
Efficiency			
Avg. travel time savings (secs)	–	521	609
Avg. number of transfers reduced	–	0.208	0.345

Table 3.2: Impact of Hubway at the network level

With larger coverage in 2012, the percentage of O-D pairs that benefit from the Hubway service is doubled to 17.74%. The increase in network efficiency also suggests that the expansion to Boston's neighboring cities helps provide better access between these areas and Boston, where major transit hubs are located. As a result, the overall network accessibility improves more substantially compared to the 2011 network. Note that while the number of stations increases by only about 55% in 2012 (from 61 to 95 stations), the number of links between bike stations within the maximum bike trip duration of 30 minutes increases by 140% (from 3,565 to 8,573 O-D pairs). This partly amplifies the impact of the Hubway network in 2012.

Node Level

In addition to the network-level evaluation, which might be obscured by a large number of O-D pairs that are not affected by the Hubway networks, we also examine the impact of Hubway at the node level. In this section, we will focus only on the Hubway network in 2012.

The changes in average travel time from a transit node to the others range from -11.4 to +1.6 minutes. The increase in average travel time of some transit nodes results from the use of the bike-sharing service to reduce the number of transfers, thereby improving the objective function value, despite the increase in travel time. In Figure 3-9, we exhibit the improvement in travel time accessibility of transit nodes

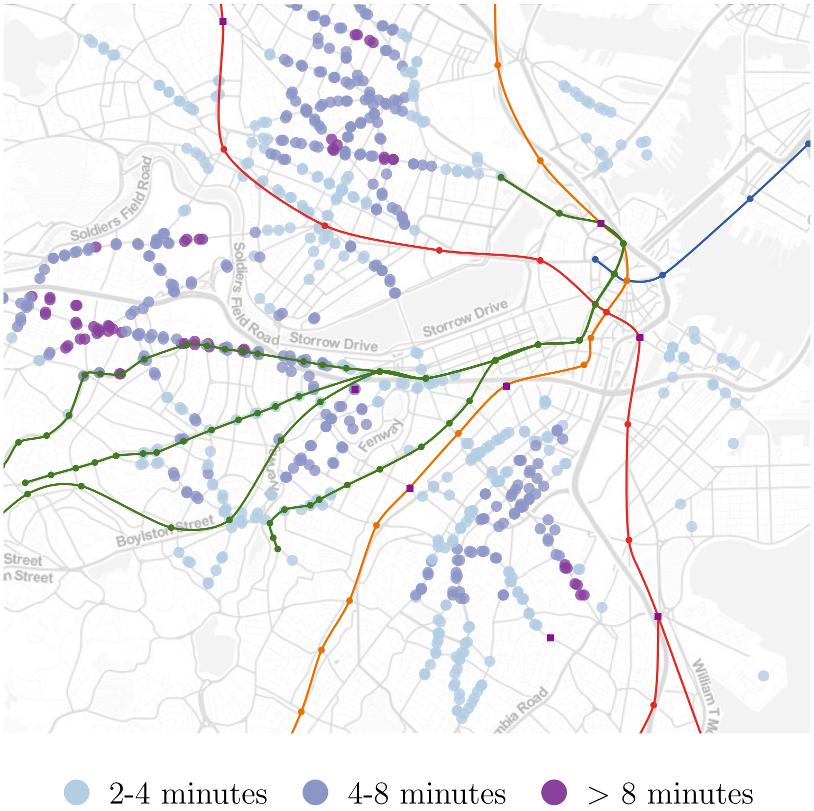


Figure 3-9: Improvement in travel time accessibility of transit nodes resulting from the Hubway's 2012 network

resulting from Hubway service in 2012. Hubway significantly improves the travel time accessibility of transit nodes that are not in close proximity to subway stations. Without Hubway service, trips originating from these transit nodes usually involve long waits for infrequent bus services and/or multiple transfers. On the other hand, travel time accessibility of transit nodes around subway stations improves minimally as they are already well connected to the existing public transit network. While omitted here, the changes in accessibility in terms of number of transfers yield similar results.

To illustrate the importance of bike stations in areas that are not well connected to the existing public transit network, we compare the accessibility and commuting patterns of two transit nodes, MIT and North Station, with and without Hubway service. For MIT, we consider the bus stop in front of MITs main entrance (77 Mass Ave). This bus stop is served by two bus services: #1 and #CT1 (whose

	MBTA	with Hubway 2012
MIT		
Avg. travel time (secs)	3,168	2,858 (-9.77%)
Avg. number of transfers	1.178	1.024 (-13.07%)
North Station		
Avg. travel time (secs)	2,508	2,480 (-1.11%)
Avg. number of transfers	0.942	0.861 (-8.59%)

Table 3.3: Accessibility of MIT and North Station

route overlaps with service #1 but skips some stops). Only service #1 is operated at a high frequency. The nearest subway station, the Kendall/MIT station, is an 8-minute walk away. North Station, on the other hand, is served by one bus, two subway, and four commuter rail services. In Hubway's 2012 network, there are bike stations, MIT (Mass Ave/Amherst St) and TD Garden, nearby the transit nodes. As shown in Table 3.3, accessibility from MIT is improved substantially compared to North Station, especially with respect to travel time.

The transfer trees associated with each scenario are provided in Figures 3-10-3-13. In Figure 3-10a, we illustrate trips originating from MIT using the MBTA network, There are seven transit nodes (including the bus stop at 77 Mass Ave) in the innermost ring, where commuters board their first transit services. Approximately 70% of commuters travel through Kendall/MIT station (Figure 3-10b). The second innermost ring, representing first transfer points in the optimal paths, is almost completely filled, indicating that most destinations (precisely, 81% of them) require at least one transfer. In the outermost ring, there are 57 destinations (less than 1%) requiring as many as four transfers.

Our interactive visualization allows users to hover the mouse cursor over a transit node to view its details, including station name, connecting services, the number of destinations whose optimal routes from the origin visit the node, optimal route to the node, and the corresponding travel time. For example, Haymarket station, shown in Figure 3-10d, is a transfer station to seven bus services serving 8% of the total set of destinations. The optimal route from MIT to Haymarket takes 25 minutes and involves walking to Kendall/MIT to use the Red line service and then transferring to an Orange line service at Downtown Crossing. One can trace this optimal path interactively as illustrated in Figures 3-10b-3-10d.

Figure 3-11a shows the transfer tree originating from MIT for the integrated MBTA and Hubway's 2012 networks. With one level fewer and minimal areas filled in the outermost ring, the resulting tree suggests a decrease in the average number of transfers. A node with an additional circle around the dot represents a bike station. As shown in Figure 3-11b, 68% of commuters now utilize the Hubway service and start their trips at the MIT (Mass Ave/Amherst St) bike station. For Haymarket station, the optimal route is to bike directly from MIT, which takes six minutes less and one transfer fewer than the previous scenario (Figure 3-11c).

For North Station, its transfer tree over the MBTA network is shown in Figure 3-12a. Most of the transfer points can be reached within 15 minutes. This explains the lower average travel time for trips originating from North Station compared to MIT, presented in Table 3.3. The majority of commuters can board their first transit services at North Station without walking to nearby transit nodes (Figure 3-12b). South Station is a transit node at which a large number of commuters beginning their trip at North Station make a transfer. Despite being two major transit hubs, North Station and South Station are not connected by subway service (Figure 3-6). The optimal route from North Station to South Station takes 13 minutes and involves a long walk (Figure 3-12c).

With Hubway service in place, commuting patterns of most trips from North Station remain unchanged, that is, most board their first transit services at North Station (as shown in Figure 3-13a). For 22% of the destinations, however, Hubway service provides a benefit, and the optimal routes for these destinations begin at the TD Garden bike station (Figure 3-13b). Hubway service establishes a direct link between North Station and South Station, providing a minute of travel time savings (Figure 3-13c). As it connects the two major transit hubs, this link accounts for the largest number of trips in the historical data. Note that we only focus on the morning peak, when most people commute to work. The frequencies of the subway service are lower during non-peak periods, translating into greater savings with Hubway.

	Station	No. of transit services	Utilization (% of O-D pairs)	Avg. travel time savings (secs)
1.	TD Garden	🚌 1 Ⓣ 2 🚆 4	3.86	419
2.	Harvard Square (Mass Ave/Dunster)*	🚌 13 Ⓣ 1	2.03	459
3.	Dudley Square	🚌 7 Ⓣ 1	2.00	789
4.	Andrew Station*	🚌 7 Ⓣ 1	1.56	515
5.	Ruggles Station	🚌 14 Ⓣ 1 🚆 3	1.22	626
6.	Somerville City Hall*	🚌 4	1.07	1,034
7.	South Station	🚌 5 Ⓣ 2 🚆 6	1.03	347
8.	Central Square*	🚌 8 Ⓣ 1	1.00	447
9.	Roxbury Crossing Station	🚌 10 Ⓣ 1	0.81	575
10.	Union Square (Somerville)*	🚌 5	0.77	834

* bike stations introduced in the 2012 network

Table 3.4: Top 10 Hubway stations with the highest utilization

In measuring the efficiency of Hubway stations, Figure 3-14 shows the top 20 stations that provide the highest average travel time savings, ranging from about 13 to 19 minutes. Their locations coincide with the areas where travel time accessibility is significantly improved.

The top 10 Hubway stations with the highest potential utilization are listed in Table 3.4. They all can be considered major transit hubs as they are served by many transit lines. However, not all of them provide large travel time savings. This suggests that while bike stations located near transit hubs, like TD Garden, might not substantially improve accessibility of neighboring transit nodes, they together with the other bike stations, like MIT, improve overall accessibility by providing better connections to major transit hubs, enabling efficient multi-modal trips over the integrated network. Given the volume of commuters traveling through transit hubs, these bike stations are not less important than the stations distant from transit hubs. Consequently, a good design of a one-way vehicle-sharing network should strive to balance the two types of stations so that the impact of integration with an existing public transit network is maximized.

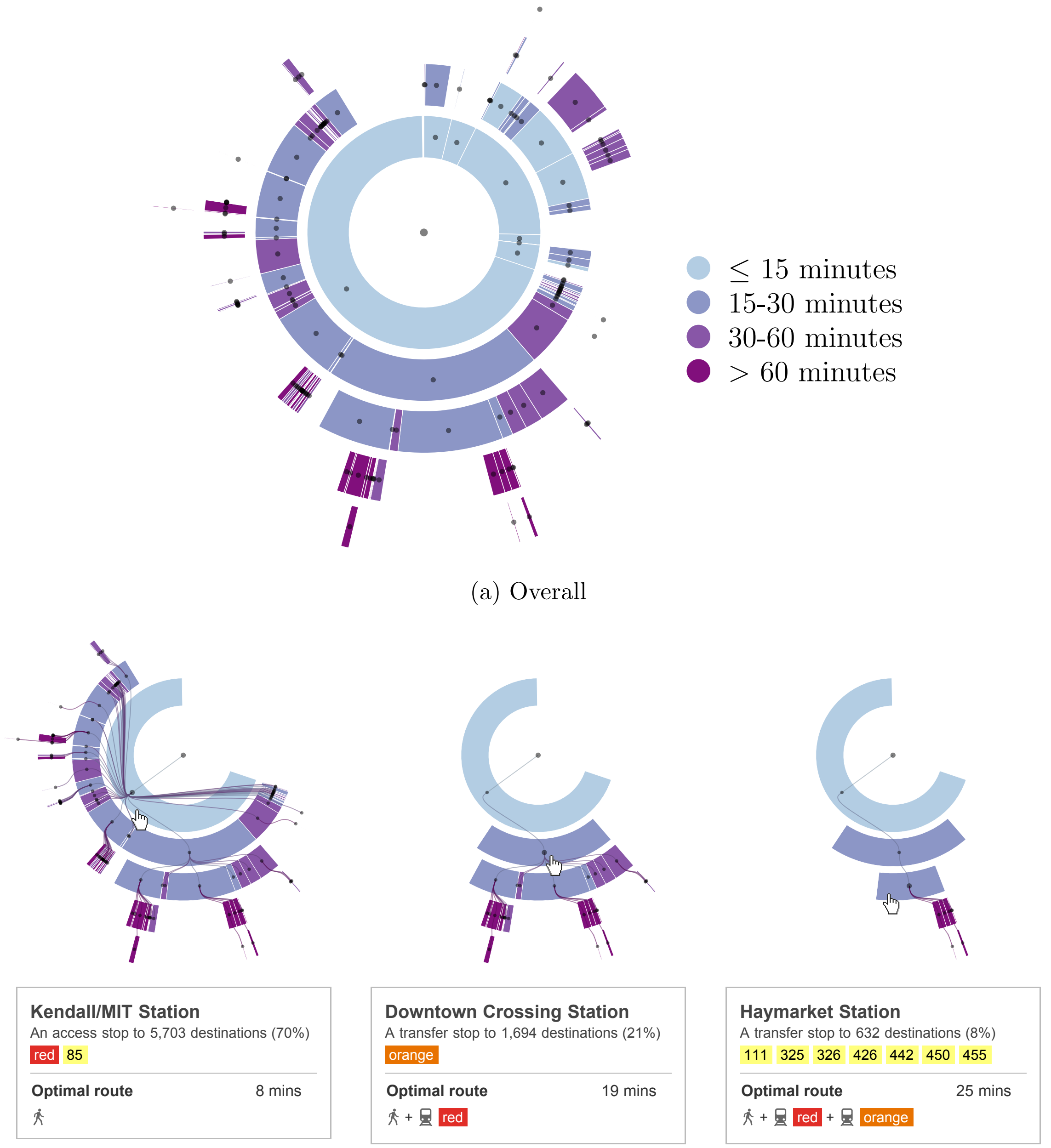
3.7 Conclusions and Future Work

In this chapter, we present a framework for evaluating the impacts of integrating one-way vehicle-sharing and public transportation services. The modeling of public transit graphs representing integrated multi-modal transportation services in this framework allows us to assess the impacts of vehicle-sharing networks at the level of O-D's. The accessibility, utilization, and efficiency metrics together can be used to measure the benefits of vehicle-sharing networks from different aspects. In addition to the evaluation metrics, we introduce the notion of a transfer tree and develop an interactive visualization tool that facilitates understanding of changes in commuting patterns resulting from vehicle-sharing services, as demonstrated in the Boston case study.

Because our framework utilizes publicly available data and web services, transportation engineers and urban planners can apply it to evaluate a potential or existing vehicle-sharing network in any city whose transit schedules are published in the increasingly adopted GTFS format. In fact, with appropriate modification, the framework can also be used to evaluate other types of transportation services that aim at complementing existing public transportation service, like shuttle bus services.

In terms of future research, we identify two interesting directions. The first direction is to incorporate travel demand information into the framework. In this work, we focus on the use of our framework for strategic planning, that is, to understand how the integrated vehicle-sharing and public transit network improves mobility in a city. We therefore treat each O-D pair equally so that the results are not dominated by O-D pairs with large demand. In fact, O-D pairs with small demand deserve special attention because it is economically infeasible for transit operators to provide them with high levels of traditional public transit services, and they are likely to benefit greatly from vehicle-sharing services. However, if the framework were to be used by vehicle-sharing operators to estimate usage, or more importantly, revenue, each O-D pair should be weighted by its travel demand to obtain more accurate estimates.

The second direction is to consider operational performance of both public transit and vehicle-sharing services that can potentially influence a commuter's mode choice. For instance, one of the advantages of cycling over taking buses is less travel time variability as bus travel times are subject to traffic conditions. On the other hand, fleet imbalances in one-way vehicle-sharing systems—having no vehicles at some stations and no parking spaces at others—can potentially limit the impacts of these services. To take these levels of services into account, additional historical operations data are required.



(a) Overall

Kendall/MIT Station
An access stop to 5,703 destinations (70%)
red 85

Optimal route 8 mins

Downtown Crossing Station
A transfer stop to 1,694 destinations (21%)
orange

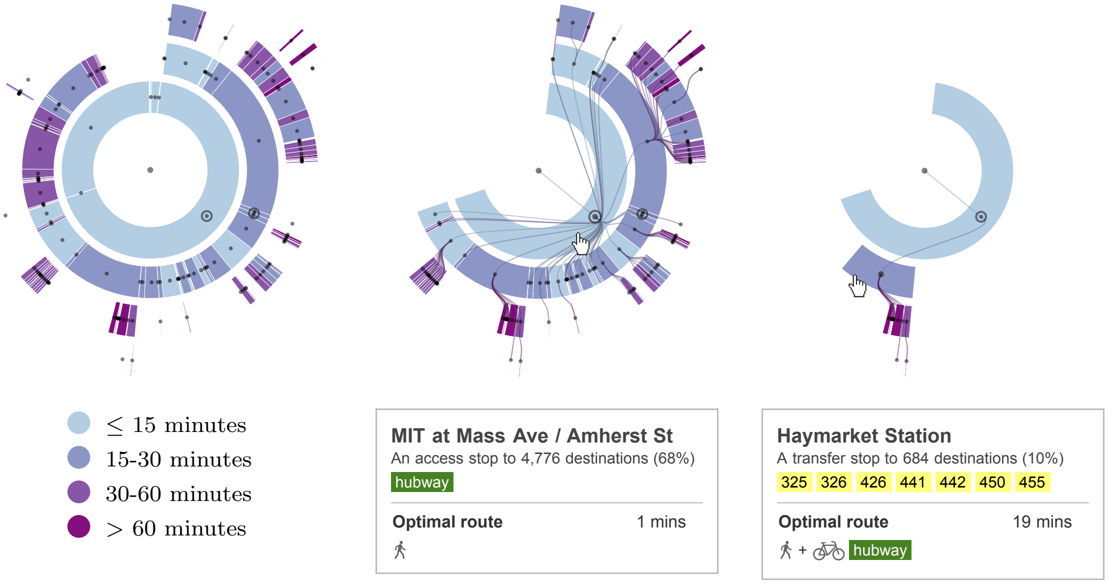
Optimal route 19 mins
+ red

Haymarket Station
A transfer stop to 632 destinations (8%)
111 325 326 426 442 450 455

Optimal route 25 mins
+ red + orange

(b) Kendall/MIT (c) Downtown Crossing (d) Haymarket

Figure 3-10: A transfer tree of trips originating from MIT over the MBTA network



MIT at Mass Ave / Amherst St
An access stop to 4,776 destinations (68%)
hubway

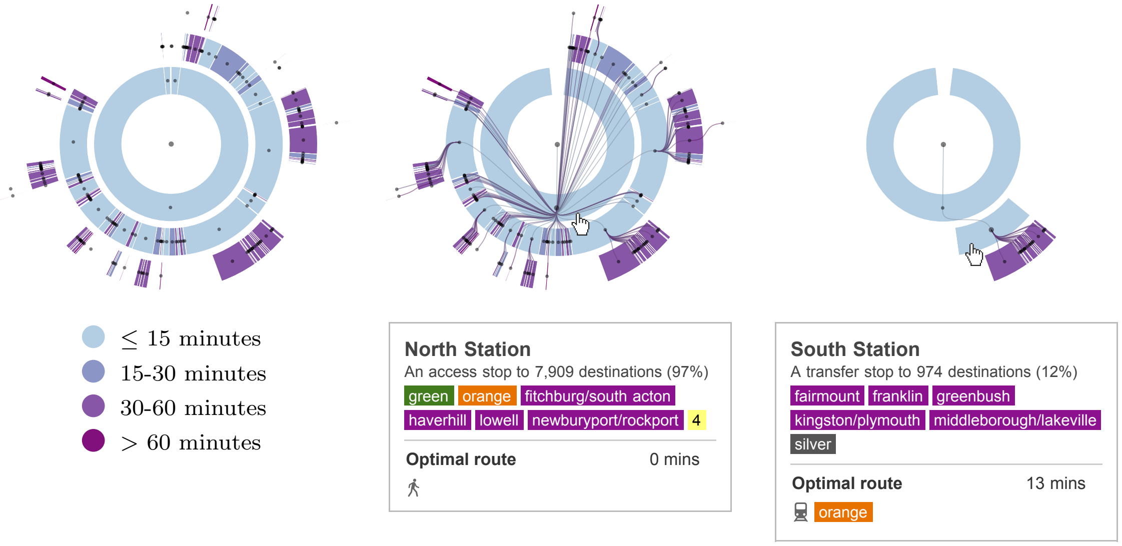
Optional route 1 mins

Haymarket Station
A transfer stop to 684 destinations (10%)
325 326 426 441 442 450 455

Optimal route 19 mins
+ hubway

(a) Overall (b) MIT (c) Haymarket

Figure 3-11: A transfer tree of trips originating from MIT over the integrated MBTA and Hubway's 2012 networks



North Station
An access stop to 7,909 destinations (97%)
green orange fitchburg/south acton haverhill lowell newburyport/rockport 4

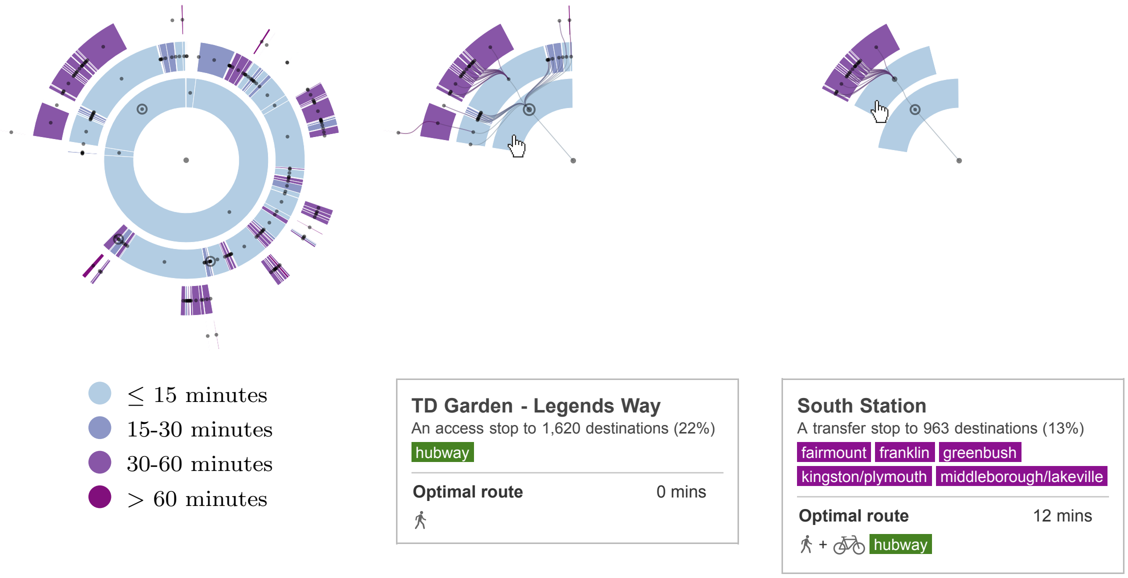
Optimal route 0 mins

South Station
A transfer stop to 974 destinations (12%)
fairmount franklin greenbush kingston/plymouth middleborough/lakeville silver

Optimal route 13 mins
orange

(a) Overall (b) North Station (c) South Station

Figure 3-12: A transfer tree of trips originating from North Station over the MBTA network



TD Garden - Legends Way
An access stop to 1,620 destinations (22%)
hubway

Optimal route 0 mins

South Station
A transfer stop to 963 destinations (13%)
fairmount franklin greenbush
kingston/plymouth middleborough/lakeville

Optimal route 12 mins
+ hubway

(a) Overall (b) North Station (c) South Station

Figure 3-13: A transfer tree of trips originating from North Station over the integrated MBTA and Hubway's 2012 networks

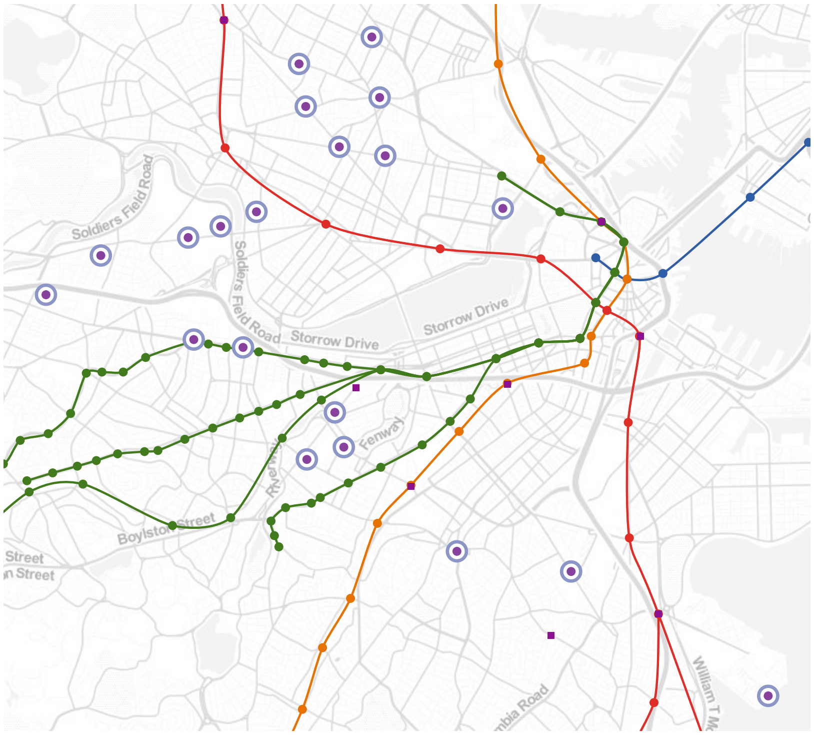


Figure 3-14: Top 20 Hubway stations with the highest efficiency

Chapter 4

Designing Integrated Vehicle-Sharing and Public Transportation Services

4.1 Introduction

As discussed in the previous chapter, one of the major benefits of one-way vehicle-sharing services is its potential integration with traditional public transportation. It can provide better access to existing public transportation and additional options for trips beyond those provided by public transit. Especially, in the areas with low travel demand, where it is economically infeasible to operate many traditional public transit services at high frequencies, on-demand vehicle-sharing services can be a more flexible, attractive alternative.

This part of the thesis concerns passenger-centric strategic planning for one-way vehicle-sharing systems. In particular, we address the vehicle-sharing network design problem in which we seek to select, from a set of candidate stations, an optimal subset of locations at which installing vehicle-sharing stations minimizes overall travel time over the integrated vehicle-sharing and public transportation network. We assume that the number of stations to be installed is predetermined, and all commuters travel on their best available routes, as opposed to making probabilistic route choices. With a proper vehicle-sharing network design, a number of commuters can reduce their travel times by using the vehicle-sharing service to better access transit hubs or travel directly to their destinations without long waits or multiple transfers which might be inevitable in the existing public transit network by itself. The literature on vehicle-sharing network design is very limited, and to our knowledge, this work is the first one that takes a passenger-centric approach to solving this problem.

The contributions of this work are as follows. We first present a mixed integer program for solving the vehicle-sharing network design problem. Because the problem consists of two sets of decisions that are made sequentially—a vehicle-sharing network design and optimal commuting paths, it is natural to use Benders decomposition as a solution approach to tackle large instances. While a tight formulation generally generates stronger Benders cuts, it requires a large number of variables and constraints, and hence, more computational effort. We propose an alternative formulation that aggregates various variables and constraints. An optimal solution to this aggregate formulation can be obtained very fast at the expense of weaker cuts, which result in more iterations required for convergence. To overcome this, we develop new algorithms that take an optimal solution to the aggregate formulation and incrementally adjust it to produce stronger Benders cuts. Consequently, this process enables us to produce strong Benders cuts quickly. As a proof of concept, we present computational results obtained using data from the Boston metropolitan area. The results confirm the effectiveness of our solution approach.

In the next section, we provide reviews of related work and the fundamentals of Benders decomposition. In Section 4.3, we state the vehicle-sharing network design problem in detail, present a mixed integer program for this problem, and describe the Benders reformulation. The alternative formulation together with the new algorithms for strengthening Benders cuts are proposed in Section 4.4. In Section 4.5, we present computational results. Finally, we conclude the work and discuss future work in Section 4.6.

4.2 Background

In this section, we survey the literature related to vehicle-sharing network design and provide a review of Benders decomposition, which is the solution approach we use in this work.

4.2.1 Related Work

Much of the research on optimization in one-way vehicle-sharing operations has been focused on fleet rebalancing operations (Kek et al., 2009; Shu et al., 2010; Nair and Miller-Hooks, 2010; Benchimol et al., 2011; Contardo et al., 2012; Raviv and Kolka, 2013; Raviv et al., 2013; Chemla et al., 2013). The literature concerning vehicle-sharing network design is very limited.

Awasthi et al. (2007, 2008) present a multi-stage decision making process for identifying locations for car-sharing stations. The process relies on ratings from experts based on several criteria. A candidate station is open if its overall score exceeds a specific threshold.

Lin and Yang (2011) propose a nonlinear integer program to determine locations of bike-sharing stations. Their objective function is to minimize the total user and operator costs. Additionally, they impose minimum service level constraints. In particular, a certain fraction of O-D pairs must have bike-sharing stations near their origins and destinations, and each station must have bicycles available with a high probability. A small hypothetical network with 11 candidate stations and 72 O-D pairs is used to test the model and study the sensitivity of parameters. While it is advantageous to incorporate operational considerations into the location problem, the model as it is presented is not sufficiently detailed to accurately account for the actual operations. Specifically, it captures only the number of bikes leaving each station, not incoming. It also assumes that bike inventory at each station can be replenished with a constant lead time, rather than considering realistic bicycle relocation operations.

Kumar and Bierlaire (2012) analyze historical trip data from Auto Bleue, a car-sharing service in Nice, and fit a linear regression model to estimate performance of car-sharing stations, measured by average daily trips. The explanatory variables associated with a given station include, among others, public transport ridership, number of nearby car-sharing stations, sizes of targeted population groups, and presence of different types of establishments, such as colleges and commercial centers, in the vicinity of the station. Some of these variables are defined based on the locations of other stations to capture the diminishing benefits of having multiple stations within the same catchment. The authors propose a nonlinear integer program to identify a subset of locations that maximizes the total expected daily trips and present a heuristic for solving the optimization problem.

The dissertation of Nair (2010) is the only work in the literature that, similarly to our work, explicitly considers integration between vehicle-sharing and public transit services. He proposes a bilevel, mixed-integer program to determine an optimal vehicle-sharing network configuration. Capturing a vehicle-sharing operator's decisions, the upper-level program determines locations of bike stations, numbers of docking slots at each station, and base inventory of vehicles at each station. It maximizes the overall utilization, subject to budget and equity constraints. Given the decisions from the upper-level program, the lower-level program determines optimal commuter flows over the integrated vehicle-sharing and public transit networks using the passenger assignment model of Spiess and Florian (1989). An exact solution method and a meta-heuristic approach for solving the optimization problem are presented. In the exact solution method, Nair exploits the convexity of the lower-level problem and transforms the bilevel program into a single mixed integer program. Using five randomly-generated networks, he demonstrates the use of the model and performs sensitivity analysis on the infrastructure cost assumptions. Despite the relatively small test networks, the model cannot be solved to optimality within an hour for many instances. In the meta-heuristic approach, Nair considers a simpler problem. In particular, station capacity and base inventory decisions are no longer included in the upper-level problem, and it is assumed that the number of stations to be installed is known a priori. This makes it very similar to the problem we consider in this work. The major difference lies in the objective function. Instead of maximizing the overall utilization (and hence, the operator's revenue), we take a passenger-centric approach and focus on minimizing the overall travel time for commuters. As shown in his results, the two objectives are not necessarily aligned. Moreover, public transit network data are preprocessed differently. Specifically, to limit problem sizes, he considers only major transit services in the existing public transit network. In contrast, as discussed in Section 4.5.1, we first determine, for each O-D pair, the optimal path over the full public transit network and then use the transfer tree concept introduced in the previous chapter to identify important transit nodes and aggregate O-D pairs based on their commuting patterns. Nair uses a genetic algorithm to solve this simplified problem for the Washington D.C. metropolitan area with 455 candidate stations and 1,000 O-D pairs. Computational time is not reported, and no discussion on solution quality is provided.

In additional to the research on vehicle-sharing network design, our problem also shares similarities with several classic problems in the operations research literature, specifically, the network design problems (Magnanti and Wong, 1984; Minoux, 1989; Kim et al., 1999) and the discrete location problems (Mirchandani and Francis, 1990; Hamacher and Drezner, 2002; Reese, 2006; Mladenović et al., 2007; Smith et al., 2009; Daskin, 2011). As described in the *general network design problems* of Contreras and Fernández (2012), these problems can be classified based on design and operational decisions. The design decision involves locating facilities or creating links between nodes, and the operational decision involves assigning customers to facilities or routing commodities through available links and/or facilities. Different types of costs may be imposed on both decisions. Table 4.1 provides a comparison between our work and some classic optimization problems.

4.2.2 Benders Decomposition

A common approach to solving large-scale (mixed) integer programs is through problem decomposition. Benders decomposition is one of the standard decomposition algorithms that has been successfully applied to solve a wide range of problems, including network design (Magnanti et al., 1986; Costa, 2005, and references therein),

Problems	Facility		Link		Assignment		Routing	
	decision	cost	decision	cost	decision	cost	decision	cost
p-median	✓				✓	✓		
Facility location	✓	✓			✓	✓		
Network design			✓	✓			✓	✓
Our problem	✓						✓	✓

Table 4.1: Comparison between the problem we address and classic optimization problems

facility/hub location (Geoffrion and Graves, 1974; de Camargo et al., 2008; Contreras et al., 2011, 2012), and integrated airline scheduling (Cordeau et al., 2001; Mercier et al., 2005; Papadakos, 2009). It is particularly attractive for solving problems that involve multi-stage decision making, or more generally, problems in which once a subset of decision variables are fixed, the optimal values of the others can be determined easily.

We now briefly explain the classical Benders decomposition algorithm (Benders, 1962) applied to a mixed integer program. Let vectors x and y represent continuous and integer decision variables, respectively. Consider a mixed integer program of the form:

$$\begin{aligned} \text{minimize} \quad & cx + dy \\ \text{subject to} \quad & Ax + Dy \geq b \\ & Fy \geq f \\ & x \in \mathbb{R}_+^{n_1},\, y \in \mathbb{Z}_+^{n_2}, \end{aligned} \tag{4.1}$$

where $c \in \mathbb{R}^{n_1}, d \in \mathbb{R}^{n_2}, b \in \mathbb{R}^{m_1}, f \in \mathbb{R}^{m_2}, A \in \mathbb{R}^{m_1 \times n_1}, D \in \mathbb{R}^{m_1 \times n_2}$, and $F \in \mathbb{R}^{m_2 \times n_2}$. We assume that there exists a *finite* optimal solution to this problem.

Let Y denote the set of feasible solutions of the variables y, $\{y \in \mathbb{Z}_+^{n_2} \mid Fy \geq f\}$. For a given $\bar{y} \in Y$, the vector x that minimizes problem (4.1) can be obtained from the following *primal subproblem* (PS)

$$\begin{aligned} z(\bar{y}) \quad = \quad \text{minimize} \quad & cx \\ \text{subject to} \quad & Ax \geq b - D\bar{y} \\ & x \in \mathbb{R}_+^{n_1}. \end{aligned} \tag{PS}$$

Let $z(\bar{y})$ equal the optimal cost of the PS for a given $\bar{y}$ if it is feasible (otherwise, $z(\bar{y}) = +\infty$). The original problem (4.1) can then be rewritten as

$$\begin{aligned} \text{minimize} \quad & z(y) + dy \\ \text{subject to} \quad & y \in Y. \end{aligned} \tag{4.2}$$

Essentially, we project the original problem (4.1) onto the space of the integer decision variables y.

By the strong duality theorem, we have that

$$z(\bar{y}) = \max_{\pi \geq 0} \{\pi(b - D\bar{y}) \mid \pi A \leq c\}, \tag{DS}$$

where π is the vector of dual variables associated with the constraint in the PS. This maximization problem is referred to as the *dual subproblem.*

Note that the feasible region Π of the DS, $\{\pi \in \mathbb{R}_+^{m_1} \mid \pi A \leq c\}$ is independent of a given $\bar{y}$. The assumption that the original problem (4.1) has a *finite* optimal solution implies that Π is *nonempty* and thus can be characterized by a set P of extreme points and a set Q of extreme rays. The assumption also implies that the optimal cost of the DS is finite (otherwise, the original problem is infeasible). It follows that

$$q(b - D\bar{y}) \leq 0, \quad \forall q \in Q, \tag{4.3}$$

that is, moving along the extreme rays does not improve the objective function value. Additionally, the optimal cost $z(\bar{y})$ can be expressed in terms of extreme points,

$\max_{p \in P} p(b - D\bar{y})$. Equivalently, $z(\bar{y})$ is the smallest number z such that

$$p(b - D\bar{y}) \leq z, \quad \forall p \in P. \tag{4.4}$$

Combining (4.2), (4.3), and (4.4), we reformulate the original problem (4.1) as the following,

$$\begin{aligned} \text{minimize} \quad & z + dy \\ \text{subject to} \quad & q(b - Dy) \leq 0 \quad \forall q \in Q \\ & p(b - Dy) \leq z \quad \forall p \in P \\ & y \in Y. \end{aligned} \tag{MP}$$

This formulation is referred to as the *master problem* (MP), and constraints (4.3) and (4.4) are called *feasibility cuts* and *optimality cuts*, respectively.

Because the number of extreme points and extreme rays of Π, and hence the number of constraints in the MP can be very large, and most of the constraints are not binding at the optimal solution anyway, the Benders decomposition algorithm uses delayed constraint generation. Specifically, the algorithm maintains the *relaxed master problem* (RMP) that contains only a subset of cuts from the MP, that is, the sets P and Q of extreme points and extreme rays in constraints (4.3) and (4.4) are replaced with some $\hat{P} \subset P$ and $\hat{Q} \subset Q$, and iteratively adds more cuts to the formulation.

The classical Benders decomposition algorithm is summarized in Algorithm 1. At iteration t, we solve the RMP with the subsets $\hat{P}$ and $\hat{Q}$ of extreme points and extreme rays for an optimal solution (y^t, z^t). Note that the optimal cost of the RMP provides a lower bound (lb) to the MP, and hence the original problem (4.1), as it is less restricted. We then solve the DS for an optimal solution π^t, given the optimal solution values y^t as parameters. If the DS is unbounded, we obtain an extreme ray and add the associated feasibility cut to the RMP. Otherwise, the DS has a finite optimal solution. We obtain an extreme point and add the associated optimality cut to the RMP. Additionally, in this case, we have that y^t is a feasible solution to the original problem (4.1), and therefore, its cost $z(y^t) + dy^t$ provides an upper bound

(ub) to the original problem (4.1).

The algorithm terminates when the lower bound is equal to the upper bound (i.e., the optimal cost of the current RMP is equal the cost of the incumbent solution), or more practically, when the gap is sufficiently small.

Algorithm 1 The classical Benders decomposition algorithm

```
$t \leftarrow 0$
$\hat{P}, \hat{Q} \leftarrow \emptyset$
$ub \leftarrow \infty,\ lb \leftarrow -\infty$
loop
    Solve the RMP with $\hat{P}$ and $\hat{Q}$ for $(y^t, z^t)$
    $lb \leftarrow z^t + dy^t$
    if $ub = lb$ then
        break
    end if
    Solve the DS with $y^t$ for $\pi^t$
    if $z(y^t) < \infty$ then
        $\hat{P} \leftarrow \hat{P} \cup \{\pi^t\}$
        if $z(y^t) + dy^t < ub$ then
            $ub \leftarrow z(y^t) + dy^t$
        end if
    else
        $\hat{Q} \leftarrow \hat{Q} \cup \{\pi^t\}$
    end if
    $t \leftarrow t + 1$
end loop
```

Depending on the underlying structure of a problem, Benders decomposition might suffer from slow convergence. A number of techniques have been proposed in the literature to accelerate the algorithm.

One computational bottleneck in the algorithm is to solve to optimality the RMP, which is an integer program, at every iteration. Because Benders cuts can be obtained from any extreme point or extreme ray of the feasible region of the DS, any feasible solution to the RMP could be used in the objective function of the DS to generate a valid cut. McDaniel and Devine (1977) propose adding a cut generated from the solution of the RMP linear program relaxation at each iteration, in addition to the integer optimal solution. Geoffrion and Graves (1974) suggest that one could stop solving the RMP once an integer feasible solution $\bar{y}$ with an objective function value smaller than $ub - \epsilon$ is found (for some nonnegative tolerance parameter ϵ), and then solve the DS with $\bar{y}$ to generate a new cut. The incumbent solution at termination is an ϵ-optimal solution to the original problem. Côté and Laughton (1984) also point out that one could iteratively generate cuts from integer feasible solutions obtained from heuristics that are computationally less expensive than the RMP. This approach however does not guarantee optimality at termination. Recently, several works (Fortz and Poss, 2009; Naoum-Sawaya and Elhedhli, 2010; Adulyasak et al., 2012) demonstrate integration of Benders decomposition and the branch-and-cut framework (see Nemhauser and Wolsey, 1988). In particular, the RMP is solved using the standard branch-and-bound method, and Benders cuts are iteratively added to the formulation at each node of the branch-and-bound tree.

When a Benders subproblem can be decomposed into smaller independent subproblems (more precisely, the coefficient matrix A associated with the variable x in the PS is block diagonal), multiple cuts associated with each independent subproblem can be generated and added to the RMP at each iteration (Birge and Louveaux, 1988), as opposed to aggregating them into a single cut. de Camargo et al. (2008) show that while this multi-cut approach is effective in reducing the number of iterations required for convergence, a large number of cuts added to the RMP make it more difficult to solve and increase the overall computation time. On the other hand, Tsamasphyrou et al. (2000), Contreras et al. (2011) and Adulyasak et al. (2012) demonstrate that, instead of adding one cut for each independent subproblem, one can aggregate the cuts to a certain level and still benefit from the reduced number of iterations.

In cases where the DS is degenerate, there are multiple optimal solutions, and each could lead to a different Bender cut. Some cuts are stronger–more effective in reducing the number of iterations—than others. Magnanti and Wong (1981) introduce the notion of *Pareto optimal* (PO) cuts, defined as follows.

Definition 1. (Magnanti and Wong, 1981)

(i) The cut $\pi^1(b - Dy) \leq z$ generated from a dual solution $\pi^1 \in \Pi$ *dominates* or *stronger* than the cut $\pi^2(b - Dy) \leq z$ generated from $\pi^2 \in \Pi$ if $\pi^1(b - Dy) \geq$

$\pi^2(b - Dy)$ for all $y \in Y$, with a strict inequality for at least one point.

(ii) A cut is *Pareto optimal* if no cut dominates it.

The authors also provide a method for obtaining Pareto optimal cuts. Let a *core point* y^0 be a point in the relative interior of the convex hull of Y. They prove that for a given $\bar{y} \in Y$, the cut generated from an optimal solution to the following *auxiliary problem* (AP) is Pareto optimal.

$$\begin{aligned} \text{maximize} \quad & \pi(b - Dy^0) \\ \text{subject to} \quad & \pi(b - D\bar{y}) = z(\bar{y}) \\ & \pi A \leq c \\ & \pi \geq 0 \end{aligned} \tag{AP}$$

The AP essentially selects, among the optimal solutions to the DS for the given $\bar{y}$, a solution that maximizes the objective function of the DS for a core point y^0. Note that different core points might result in different PO cuts.

Geoffrion and Graves (1974) and Magnanti and Wong (1981) discuss the impact of different formulations of a mixed integer program on convergence of the Benders decomposition algorithm. In particular, despite a large number of constraints, a *tighter* formulation—one whose linear programming (LP) relaxation has a smaller feasible region—is more desirable as it provides stronger Benders cuts.

4.3 Vehicle-Sharing Network Design Problem

Consider a directed graph $G = (\mathcal{N}, \mathcal{A})$ consists of a set $\mathcal{N}$ of nodes and a set $\mathcal{A}$ of arcs representing an existing public transit network together with a potential vehicle-sharing (VS) network. Nodes in set $\tilde{\mathcal{N}} \subset \mathcal{N}$ are designated as candidate locations for VS stations. Starting and ending at candidate VS stations, arcs in set $\tilde{\mathcal{A}} \subset \mathcal{A}$ represent VS service and are available only if both stations are installed. Let $\mathcal{H} \subseteq \mathcal{N} \times \mathcal{N}$ denote a set of origin-destination (O-D) pairs. Associated with each O-D pair

$h \in \mathcal{H}$ is a travel demand w^h from the origin $o(h)$ to the destination $d(h)$. Traveling along arc $(i, j) \in \mathcal{A}$ incurs a positive cost c_{ij}. Note that the cost is independent of a commuter's origin and destination. Lastly, to satisfy an operator's budget constraint, at most K stations can be installed.

There are two sets of decisions to be made in this problem: (1) a design of a VS network specifying locations of VS stations to be installed; and (2) for each O-D pair, a commuting path that minimizes the total travel cost.

4.3.1 Mixed Integer Program Formulation

Let x_{ij}^h denote the number of commuters of O-D pair $h \in \mathcal{H}$ traveling on arc $(i, j) \in \mathcal{A}$, and y_i be a binary decision variable that equals one if a VS station is installed at node $i \in \tilde{\mathcal{N}}$. A mixed integer program for the vehicle-sharing network design problem (VSND) can be formulated as follows.

$$\text{minimize} \qquad \sum_{h \in \mathcal{H}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^h \tag{4.5}$$

subject to

$$\sum_{(i,j) \in \mathcal{A}} x_{ij}^h - \sum_{(j,i) \in \mathcal{A}} x_{ji}^h = \begin{cases} w^h & i = o(h) \\ -w^h & i = d(h) \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in \mathcal{H},\ \forall i \in \mathcal{N} \tag{4.6}$$

$$x_{ij}^h \le w^h y_i \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.7}$$

$$x_{ij}^h \le w^h y_j \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.8}$$

$$\sum_{i \in \tilde{\mathcal{N}}} y_i \le K \tag{4.9}$$

$$x_{ij}^h \ge 0 \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \mathcal{A} \tag{4.10}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \tilde{\mathcal{N}} \tag{4.11}$$

The objective (4.5) is to minimize the total travel cost of commuters assigned to all arcs. The flow of commuters of each O-D pair is conserved at each node by constraint (4.6). Constraints (4.7) and (4.8) ensure that commuters cannot use the VS service between nodes i and j unless both stations are installed. Lastly, constraint (4.9) limits the number of stations installed to K stations.

Because the arcs representing existing public transit service in the set $\mathcal{A} \setminus \tilde{\mathcal{A}}$ are always included in the graph, this problem is feasible for any VS network design. Additionally, given that the arc costs are positive, the cost of the optimal design is finite.

While the formulation is quite simple, the numbers of variables and constraints grow substantially as the network size and/or the number of O-D pairs increase. This potentially prohibits the use of this formulation to solve problems representing real-world urban transportation networks.

4.3.2 Benders Reformulation

Because the VSND consists of two sets of decisions that are made sequentially—a VS network design and optimal commuting paths, it is intuitive to tackle large instances of the VSND using Benders decomposition, outlined in Section 4.2.2.

For a given VS network design $\bar{y}$, we can solve the following *primal subproblem* (PS) for the optimal commuting paths.

$$\text{minimize} \qquad \sum_{h \in \mathcal{H}} \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^h \tag{4.12}$$

subject to

$$\sum_{(i,j) \in \mathcal{A}} x_{ij}^h - \sum_{(j,i) \in \mathcal{A}} x_{ji}^h = \begin{cases} w^h & i = o(h) \\ -w^h & i = d(h) \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in \mathcal{H},\ \forall i \in \mathcal{N} \tag{4.13}$$

$$x_{ij}^h \leq w^h \bar{y}_i \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.14}$$

$$x_{ij}^h \leq w^h \bar{y}_j \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.15}$$

$$x_{ij}^h \geq 0 \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \mathcal{A} \tag{4.16}$$

Because there is no capacity imposed on each arc *included* in the network, all commuters travel on their shortest available paths in the optimal solution. Consequently, this problem reduces to a collection of $|\mathcal{H}|$ shortest path problems, one for each O-D pair $h \in \mathcal{H}$, over the set of available arcs, $\mathcal{A} \setminus \{(i,j) \in \tilde{\mathcal{A}} \mid \bar{y}_i = 0 \vee \bar{y}_j = 0\}$.

Let p_i^h, u_{ij}^h, and v_{ij}^h denote the negatives of the dual variables associated with constraints (4.13), (4.14), and (4.15), respectively. The corresponding *dual subproblem* (DS) is given by:

$$\begin{aligned} \text{maximize} \qquad z(\bar{y}) = \sum_{h \in \mathcal{H}} w^h \Bigg[& \left(p_{d(h)}^h - p_{o(h)}^h\right) \\ & - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u_{ij}^h + \sum_{(j,i) \in \tilde{\mathcal{A}}} v_{ji}^h \right) \bar{y}_i \Bigg] \end{aligned} \tag{4.17}$$

subject to

$$p_j^h - p_i^h \leq c_{ij} \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \mathcal{A} \setminus \tilde{\mathcal{A}} \tag{4.18}$$

$$p_j^h - p_i^h \leq c_{ij} + u_{ij}^h + v_{ij}^h \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.19}$$

$$u_{ij}^h, v_{ij}^h \geq 0 \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}}. \tag{4.20}$$

Again, this problem can be decomposed further into $|\mathcal{H}|$ independent subproblems. Because u_{ij}^h and v_{ij}^h are the dual variables associated with constraints (4.14) and (4.15), we can interpret their values as the potential savings for O-D pair h resulting from making arc (i,j) available. Hence, the term $\left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u_{ij}^h + \sum_{(j,i)\in\tilde{\mathcal{A}}} v_{ji}^h\right)$ associated with candidate station $i \in \tilde{\mathcal{N}}$ in the objective function (4.17) represents the potential savings for O-D pair h resulting from installing station i in the VS network.

Because the underlying network structure guarantees feasibility and finite optimum of the PS for any feasible solution $\bar{y}$, by the strong duality theorem, the DS is also feasible and has a finite optimal cost. Consequently, the optimal cost of the DS is given by

$$z(\bar{y}) = \max_{(p,u,v)\in P} \sum_{h\in\mathcal{H}} w^h \left[\left(p^h_{d(h)} - p^h_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^h_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^h_{ji} \right) \bar{y}_i \right],$$

where P denotes the set of extreme points associated with the feasible region of the DS.

The Benders *master problem* (MP) can then be formulated as follows:

$$\text{minimize} \qquad z \tag{4.21}$$

subject to

$$z \geq \sum_{h\in\mathcal{H}} w^h \left[\left(p^h_{d(h)} - p^h_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^h_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^h_{ji} \right) y_i \right] \tag{4.22}$$

$$\forall (p,u,v) \in P$$

$$\sum_{i\in\tilde{\mathcal{N}}} y_i \leq K \tag{4.23}$$

$$y_i \in \{0,1\} \qquad \forall i \in \tilde{\mathcal{N}} \tag{4.24}$$

Notice that there is no Benders feasibility cut in this formulation as the DS is guaranteed to have a finite optimal cost. Because the DS can be decomposed into $|\mathcal{H}|$ independent subproblems, multiple cuts associated with each subproblem can be added to the formulation instead of a single aggregate cut. In particular, we can replace constraint (4.22) with

$$z^h \geq w^h \left[\left(p^h_{d(h)} - p^h_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^h_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^h_{ji} \right) y_i \right], \ \forall h \in \mathcal{H}, \ \forall (p,u,v) \in P$$

and change the objective function to minimize $\sum_{h\in\mathcal{H}} z^h$. We refer to this alternate formulation as *multi-cut formulation.*

Applying the classical Benders decomposition algorithm (see Algorithm 1), we solve repeatedly the *restricted master problem* (RMP) containing only a subset $\hat{P}$ of all extreme points. At each iteration, we obtain a new extreme point by solving the DS with the current optimal solution to the RMP as parameter, and add it to the set $\hat{P}$. The algorithm terminates when the optimal cost of the RMP is equal to the cost of the incumbent solution.

4.3.3 Solving Benders Subproblems

Because the DS has to be solved at every iteration to generate a new Benders cut, an efficient method for solving the DS can improve the overall computational time. While it is already straightforward to solve the linear program (4.17)-(4.20) to obtain an optimal solution (p, u, v), we can exploit the network structure of its primal (4.12)-(4.16), which can be solved more efficiently using a specialized algorithm like the network simplex algorithm. Therefore, we solve the PS as a network flow problem and obtain an optimal solution (p, u, v) to the DS from the dual values.

As mentioned earlier, the optimal solution to the PS for a given O-D pair can actually be obtained by simply finding the shortest path from the origin to the destination over the set of available arcs. We however need to solve it as a network flow problem in order to obtain a complete set of dual values that constitute an optimal solution (p, u, v) to the DS, from which we generate a Benders cut.

Pareto Optimal Cuts

When the DS has multiple optimal solutions, we can strategically select an optimal solution that corresponds to a *stronger* Benders cut. *Pareto optimal* (PO) cuts (Magnanti and Wong, 1981) are non-dominated cuts (see Definition 1) that are proven to be effective in accelerating convergence of the Benders decomposition algorithm. Let Y denote the set of feasible solutions of the variables y, $\{y \in \{0,1\}^{|\tilde{\mathcal{N}}|} \mid \sum_{i \in \tilde{\mathcal{N}}} y_i \leq K\}$, and y^0 denote a *core point*—a point in the relative interior of the convex hull of Y. As explained in Section 4.2.2, we can select, among the optimal solutions to the DS for a given $\bar{y} \in Y$, a solution that maximizes the objective function of the DS for a core point y^0. In particular, a PO cut can be generated from an optimal solution to the following *auxiliary problem* (AP):

$$\text{maximize} \qquad \sum_{h\in\mathcal{H}} w^h \left[\left(p^h_{d(h)} - p^h_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^h_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^h_{ji} \right) y^0_i \right] \tag{4.25}$$

subject to

$$z^h(\bar{y}) = w^h \left[\left(p^h_{d(h)} - p^h_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^h_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^h_{ji} \right) \bar{y}_i \right] \qquad \forall h \in \mathcal{H} \tag{4.26}$$

$$p^h_j - p^h_i \le c_{ij} \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \mathcal{A} \setminus \tilde{\mathcal{A}} \tag{4.27}$$

$$p^h_j - p^h_i \le c_{ij} + u^h_{ij} + v^h_{ij} \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.28}$$

$$u^h_{ij}, v^h_{ij} \ge 0 \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}}, \tag{4.29}$$

where $z^h(\bar{y})$ denotes the optimal cost of the DS for the given $\bar{y}$ associated with O-D pair h.

Let x^h_0 denote the negatives of the dual variables associated with constraints (4.26), and x^h_{ij} denote the dual variables associated with constraints (4.27) and (4.28). The dual of the AP is given by:

$$\text{minimize} \qquad \sum_{h\in\mathcal{H}} \left[\sum_{(i,j)\in\mathcal{A}} c_{ij} x^h_{ij} - z^h(\bar{y}) x^h_0 \right] \tag{4.30}$$

subject to

$$\sum_{(i,j)\in\mathcal{A}} x^h_{ij} - \sum_{(j,i)\in\mathcal{A}} x^h_{ji} = \begin{cases} w^h(1 + x^h_0) & i = o(h) \\ -w^h(1 + x^h_0) & i = d(h) \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in \mathcal{H},\ \forall i \in \mathcal{N} \tag{4.31}$$

$$x^h_{ij} \le w^h(y^0_i + x^h_0 \bar{y}_i) \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.32}$$

$$x^h_{ij} \le w^h(y^0_j + x^h_0 \bar{y}_j) \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.33}$$

$$x^h_{ij} \ge 0 \qquad \forall h \in \mathcal{H},\ \forall (i,j) \in \mathcal{A}. \tag{4.34}$$

As discussed in Magnanti et al. (1986), this problem could be viewed as $|\mathcal{H}|$ network flow problems each parameterized by x_0^h. Moreover, an optimal solution to each subproblem can be obtained by fixing the value of x_0^h to a sufficiently large number. Specifically, notice that each unit of additional portion of demand x_0^h decreases the objective value by $z^h(\bar{y})$. Therefore, the objective value can be improved, by increasing x_0^h, as long as the marginal cost of sending the additional flow is smaller than $z^h(\bar{y})$. Additionally, because the capacity of each arc *unavailable* in the current solution $\bar{y}$ is $w^h \min\{y_i^0, y_j^0\}$, the maximum amount of flow possibly routed through paths including these unavailable arcs is equal to $\sum_{(i,j)\in\tilde{\mathcal{A}}} w^h \min\{y_i^0, y_j^0\}$. The excess flow must be routed though the optimal path for the current network $\bar{y}$, whose cost is $z^h(\bar{y})$, and hence, increasing x_0^h beyond $\sum_{(i,j)\in\tilde{\mathcal{A}}} \min\{y_i^0, y_j^0\}$ will no longer improve the objective value.

Consequently, we can solve $|\mathcal{H}|$ network flow problems with appropriate values of x_0^h and generate a PO cut from the optimal dual values. Note that it is computationally more expensive to generate a PO cut as we first need to obtain the optimal cost of the DS for the given $\bar{y} \in Y$.

4.4 Alternate Formulation and Implementation

4.4.1 Tree Formulation

In this section, we discuss an alternative formulation that has significantly fewer decision variables and constraints and can potentially be solved more easily, especially on a machine with smaller memory. Let $\mathcal{L}$ denote the set of all origins $\cup_{h\in\mathcal{H}}\{o(h)\}$; W^l denote the total demand of all O-D pairs originating from node $l \in \mathcal{L}$; and x_{ij}^l denote the number of commuters originating from node $l \in \mathcal{L}$ and traveling on arc $(i,j) \in \mathcal{A}$. The VSND can be refomulated as follows:

$$\text{minimize} \qquad \sum_{l\in\mathcal{L}} \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}^l \tag{4.35}$$

subject to

$$\sum_{(i,j)\in\mathcal{A}} x^l_{ij} - \sum_{(j,i)\in\mathcal{A}} x^l_{ji} = \begin{cases} W^l & i = l \\ -w^h & (l,i) = h \in \mathcal{H} \\ 0 & \text{otherwise} \end{cases} \qquad \forall l \in \mathcal{L},\ \forall i \in \mathcal{N} \tag{4.36}$$

$$x^l_{ij} \le W^l y_i \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.37}$$

$$x^l_{ij} \le W^l y_j \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.38}$$

$$\sum_{i\in\tilde{\mathcal{N}}} y_i \le K \tag{4.39}$$

$$x^l_{ij} \ge 0 \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \mathcal{A} \tag{4.40}$$

$$y_i \in \{0,1\} \qquad \forall i \in \tilde{\mathcal{N}}. \tag{4.41}$$

We refer to this compact formulation as a *tree formulation.* Because the capacity constraints (4.37) and (4.38) are aggregated over O-D pairs with the same origins, the feasible region of its LP relaxation is not as tight as that of the *O-D formulation* (4.5)-(4.11). As a result, the optimal cost of its LP relaxation does not provide a good lower bound for the original mixed integer program, making it difficult to solve using standard approaches like the branch-and-bound algorithm, in which good lower bounds are required for pruning the branch-and-bound tree. While the tree formulation is less favorable in theory, as we will show shortly, it can be solved, using off-the-shelf solvers, faster than the O-D formulation, especially for large instances.

For the Benders decomposition algorithm, the tree-based PS for a given VS network design $\bar{y}$ is formulated as follows:

$$\text{minimize} \qquad \sum_{l\in\mathcal{L}} \sum_{(i,j)\in\mathcal{A}} c_{ij} x^l_{ij} \tag{4.42}$$

subject to

$$\sum_{(i,j)\in\mathcal{A}} x_{ij}^l - \sum_{(j,i)\in\mathcal{A}} x_{ji}^l = \begin{cases} W^l & i = l \\ -w^h & (l,i) = h \in \mathcal{H} \\ 0 & \text{otherwise} \end{cases} \qquad \forall l \in \mathcal{L},\ \forall i \in \mathcal{N} \tag{4.43}$$

$$x_{ij}^l \leq W^l \bar{y}_i \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.44}$$

$$x_{ij}^l \leq W^l \bar{y}_j \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.45}$$

$$x_{ij}^l \geq 0 \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \mathcal{A}. \tag{4.46}$$

The problem can be decomposed into $|\mathcal{L}|$ independent subproblems. Because the commuters travel on their shortest available paths in the optimal solution, for each subproblem associated with an origin $l \in \mathcal{L}$, we can use a standard algorithm like Dijkstra's algorithm to construct a shortest path tree rooted at node l using only arcs available in $\bar{y}$ and obtain the respective shortest paths for each destination.

Let p_i^l, u_{ij}^l, and v_{ij}^l denote the negatives of the dual variables associated with constraints (4.43), (4.44), and (4.45), respectively. The corresponding tree-based DS is given by:

$$\begin{aligned} \text{maximize} \qquad z(\bar{y}) = \sum_{l\in L} \Bigg[& \sum_{h\in\mathcal{H}^l} w^h \left(p_{d(h)}^l - p_{o(h)}^l\right) \\ & - \sum_{i\in\tilde{\mathcal{N}}} W^l \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u_{ij}^l + \sum_{(j,i)\in\tilde{\mathcal{A}}} v_{ji}^l \right) \bar{y}_i \Bigg] \end{aligned} \tag{4.47}$$

subject to

$$p_j^l - p_i^l \leq c_{ij} \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \mathcal{A} \setminus \tilde{\mathcal{A}} \tag{4.48}$$

$$p_j^l - p_i^l \leq c_{ij} + u_{ij}^l + v_{ij}^l \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.49}$$

$$u_{ij}^l, v_{ij}^l \geq 0 \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}}, \tag{4.50}$$

where $\mathcal{H}^l$ denotes the set of O-D pairs originating from node $l \in \mathcal{L}$.

As previously discussed in Geoffrion and Graves (1974) and Magnanti and Wong

(1981), a tight formulation, usually achieved through a large number of constraints, yields stronger Benders cuts. To see this in our context, recall that the term $\left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^l_{ji}\right)$ associated with candidate station $i \in \tilde{\mathcal{N}}$ can be interpreted as the potential savings for the O-D pairs originating from node l resulting from installing station i in the VS network. Because the term is weighted by the total demand W^l in the objective function, even though some of the O-D pairs might not actually benefit from station i, the total savings tend to be overestimated, and the resulting cut is not strong. We will revisit this issue and present an approach to strengthening the cuts obtained from the tree formulation in Section 4.4.2.

The tree formulation of the AP for generating a PO cut is given by

$$\text{maximize} \quad \sum_{l\in\mathcal{L}} \left[\sum_{h\in\mathcal{H}^l} w^h \left(p^l_{d(h)} - p^l_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} W^l \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^l_{ji} \right) y^0_i \right] \tag{4.51}$$

subject to

$$z^l(\bar{y}) = \sum_{l\in\mathcal{L}} \left[\sum_{h\in\mathcal{H}^l} w^h \left(p^l_{d(h)} - p^l_{o(h)}\right) - \sum_{i\in\tilde{\mathcal{N}}} W^l \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^l_{ji} \right) \bar{y}_i \right] \qquad \forall l \in \mathcal{L} \tag{4.52}$$

$$p^l_j - p^l_i \le c_{ij} \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \mathcal{A} \setminus \tilde{\mathcal{A}} \tag{4.53}$$

$$p^l_j - p^l_i \le c_{ij} + u^l_{ij} + v^l_{ij} \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.54}$$

$$u^l_{ij}, v^l_{ij} \ge 0 \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}}, \tag{4.55}$$

where $z^l(\bar{y})$ denotes the optimal cost of the DS associated with O-D pairs originating from node l, for a given $\bar{y} \in Y$.

Let x^l_0 denote the negatives of the dual variables associated with constraints (4.52), and x^l_{ij} denote the dual variables associated with constraints (4.53) and (4.54). The dual of the AP is given by:

$$\text{minimize} \qquad \sum_{l \in \mathcal{L}} \left[\sum_{(i,j) \in \mathcal{A}} c_{ij} x^l_{ij} - z^l(\bar{y}) x^l_0 \right] \tag{4.56}$$

subject to

$$\sum_{(i,j) \in \mathcal{A}} x^l_{ij} - \sum_{(j,i) \in \mathcal{A}} x^l_{ji} = \begin{cases} W^l(1 + x^l_0) & i = l \\ -w^h(1 + x^l_0) & (l,i) = h \in \mathcal{H} \\ 0 & \text{otherwise} \end{cases} \qquad \forall l \in \mathcal{L},\ \forall i \in \mathcal{N} \tag{4.57}$$

$$x^l_{ij} \leq W^l(y^0_i + x^l_0 \bar{y}_i) \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.58}$$

$$x^l_{ij} \leq W^l(y^0_j + x^l_0 \bar{y}_j) \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \tilde{\mathcal{A}} \tag{4.59}$$

$$x^l_{ij} \geq 0 \qquad \forall l \in \mathcal{L},\ \forall (i,j) \in \mathcal{A}. \tag{4.60}$$

Similarly to the O-D-based formulation, we can solve $|\mathcal{L}|$ network flow problems with appropriate values of x^l_0 and generate a PO cut from the optimal dual values.

4.4.2 Strengthening Tree Cuts

While Benders cuts generated from the O-D formulation are effective in reducing the number of iterations required for convergence, it is very time-consuming to solve one optimization problem (or two to generate PO cuts) for each O-D pair. On the other hand, through grouping O-D pairs with a common origin, the tree formulation allows us to solve fewer subproblems at each iteration at the expense of weaker cuts, which result in more iterations. In this section, we present efficient algorithms for strengthening cuts generated from the tree formulation, referred to as *tree cuts*.

Consider a feasible solution $\bar{y} \in Y$ and an origin $l \in \mathcal{L}$. Let (p^l, u^l, v^l) be an optimal solution to the subproblem of the tree-based DS (4.47)-(4.50) associated with origin l for the given $\bar{y}$. We first establish the following proposition.

Proposition 1. *An optimal solution (p^l, u^l, v^l) is also an optimal solution to the subproblem of the O-D-based DS associated with any O-D pair $h \in \mathcal{H}^l$.*

Proof. Because constraints (4.48) - (4.50) in the tree formulation are of the same form as constraints (4.18) - (4.20) in the O-D formulation, a feasible solution (p^l, u^l, v^l) to the tree formulation is also a feasible solution to the subproblem of the O-D-based DS associated with any O-D pair $h \in \mathcal{H}^l$. For each O-D pair $h \in \mathcal{H}^l$, the objective value of (p^l, u^l, v^l) is given by

$$w^h \left[\left(p^l_{d(h)} - p^l_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i) \in \tilde{\mathcal{A}}} v^l_{ji} \right) \bar{y}_i \right] \leq z^h(\bar{y}), \tag{4.61}$$

where $z^h(\bar{y})$ denotes the optimal cost of the O-D-based DS for the given $\bar{y}$ associated with O-D pair h. Summing the objective values over all O-D pairs in $\mathcal{H}^l$, we have

$$\begin{aligned}
\sum_{h \in \mathcal{H}^l} z^h(\bar{y}) &\geq \sum_{h \in \mathcal{H}^l} w^h \left[\left(p^l_{d(h)} - p^l_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i) \in \tilde{\mathcal{A}}} v^l_{ji} \right) \bar{y}_i \right] \\
&= \sum_{h \in \mathcal{H}^l} w^h \left(p^l_{d(h)} - p^l_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} W^l \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u^l_{ij} + \sum_{(j,i) \in \tilde{\mathcal{A}}} v^l_{ji} \right) \bar{y}_i \\
&= z^l(\bar{y}),
\end{aligned}$$

where $z^l(\bar{y})$ denote the optimal cost of the DS for the given $\bar{y}$ associated with O-D pairs originating from node l. Because, by definition, $\sum_{h \in \mathcal{H}^l} z^h(\bar{y})$ is equal to $z^l(\bar{y})$, it follows that inequality (4.61) holds with equality for every O-D pair $h \in \mathcal{H}^h$. □

Given this fact, our algorithms start with an optimal solution (p^l, u^l, v^l) to the tree formulation and incrementally adjust it to produce an optimal solution (p^h, u^h, v^h) to the O-D formulation from which a potentially stronger Benders cut can be generated.

Consider Benders cuts associated with O-D pair $h \in \mathcal{H}$. From Definition 1, the cut generated from (p^1, u^1, v^1) dominates or stronger than the cut generated from

(p^2, u^2, v^2) if

$$\begin{aligned} &w^h \left[\left(p^1_{d(h)} - p^1_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u^1_{ij} + \sum_{(j,i) \in \tilde{\mathcal{A}}} v^1_{ji} \right) y_i \right] \\ &\quad \geq w^h \left[\left(p^2_{d(h)} - p^2_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j) \in \tilde{\mathcal{A}}} u^2_{ij} + \sum_{(j,i) \in \tilde{\mathcal{A}}} v^2_{ji} \right) y_i \right], \end{aligned}$$

for all $y \in Y$ with a strict inequality for at least one point. Therefore, we can potentially strengthen a cut by decreasing the values of u^h and v^h, while maintaining the feasibility and optimality of (p^h, u^h, v^h). Recall our earlier discussion that Benders cuts generated from the tree formulation are not strong because the potential savings u^l_{ij} and v^l_{ij} resulting from making arc (i, j) available are counted for every O-D pair in $\mathcal{H}^l$ (i.e., weighted by W^l). By decreasing the values of u^h and v^h for each individual O-D pair $h \in \mathcal{H}^l$, we essentially make the estimate of the potential savings more accurate.

Because, for each arc $(i, j) \in \tilde{\mathcal{A}}$, the constraint $p^h_j - p^h_i \leq c_{ij} + u^h_{ij} + v^h_{ij}$ must be satisfied, decreasing p^h_j might enable us to decrease u^h_{ij} and v^h_{ij} further. Note that for an O-D pair h, besides u^h and v^h, only $p^h_{o(h)}$ and $p^h_{d(h)}$ associated with the origin $o(h)$ and the destination $d(h)$ are included the objective function. Hence, we can adjust the values of p^h associated with the other nodes without affecting the objective value.

Consequently, for an O-D pair $h \in \mathcal{H}^l$, our algorithms fix the values of $p^h_{o(h)}$ and $p^h_{d(h)}$ to their initial values $p^l_{o(h)}$ and $p^l_{d(h)}$, and incrementally *decrease* the values of u^h, v^h, and p^h associated with the other nodes; while maintaining the feasibility of (p^h, u^h, v^h). The following result ensures the optimality of (p^h, u^h, v^h), and thus, establishes the correctness of our algorithms.

Proposition 2. *Let (p^1, u^1, v^1) be an optimal solution to the O-D-based DS associated with O-D pair $h \in \mathcal{H}$ for a given $\bar{y} \in Y$, with an optimal cost of $z^h(\bar{y})$. And let (p^2, u^2, v^2) be another feasible solution. If $p^2_{o(h)} = p^1_{o(h)}, p^2_{d(h)} = p^1_{d(h)}, u^2_{ij} \leq u^1_{ij}$ for all $(i, j) \in \tilde{\mathcal{A}}$, and $v^2_{ij} \leq v^1_{ij}$ for all $(i, j) \in \tilde{\mathcal{A}}$, (p^2, u^2, v^2) is also optimal.*

Proof. From the optimality of (p^1, u^1, v^1) and the properties of (p^2, u^2, v^2), we have that

$$z^h(\bar{y}) \geq w^h \left[\left(p^2_{d(h)} - p^2_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^2_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^2_{ji} \right) \bar{y}_i \right] \tag{4.62}$$

$$= w^h \left[\left(p^1_{d(h)} - p^1_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^2_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^2_{ji} \right) \bar{y}_i \right] \tag{4.63}$$

$$\geq w^h \left[\left(p^1_{d(h)} - p^1_{o(h)}\right) - \sum_{i \in \tilde{\mathcal{N}}} \left(\sum_{(i,j)\in\tilde{\mathcal{A}}} u^1_{ij} + \sum_{(j,i)\in\tilde{\mathcal{A}}} v^1_{ji} \right) \bar{y}_i \right] \tag{4.64}$$

$$= z^h(\bar{y}). \tag{4.65}$$

Consequently, the inequalities hold with equality, and (p^2, u^2, v^2) must be optimal. □

More importantly, if u^2_{ij} (or v^2_{ij}) is strictly less than u^1_{ij} (or v^1_{ij}) for some arc $(i, j) \in \tilde{\mathcal{A}}$, it follows that the cut generated from (p^2, u^2, v^2) dominates the cut generated from (p^1, u^1, v^1).

Finally, because, by construction, an optimal solution to the tree-based AP (4.51)-(4.55) is also an optimal solution to the tree-based DS (4.47)-(4.50), our algorithms can also be used to strengthen PO tree cuts.

Using the observations and results established in this section, we now present the details of the algorithms for strengthening tree cuts.

Basic Algorithm

Consider an O-D pair $h \in \mathcal{H}^l$. The algorithm first initializes (p^h, u^h, v^h) to the associated solution to the tree formulation (p^l, u^l, v^l). For brevity of notation, we assume in this section that there are also $u^h_{ij} = 0$ and $v^h_{ij} = 0$ associated with each arc $(i, j) \in \mathcal{A} \setminus \tilde{\mathcal{A}}$. Consequently, the constraints associated with each $(i, j) \in \mathcal{A}$ are of the same form: $p^h_j - p^h_i \leq c_{ij} + u^h_{ij} + v^h_{ij}$.

Updating p^h_i. For each node $i \neq o(h)$ and $d(h)$, the algorithm decreases p^h_i to its lower bound, which is determined by the constraints associated with its outgoing arcs. In particular, to preserve feasibility, the following constraints must be satisfied

$$p_i^h \geq p_j^h - c_{ij} - u_{ij}^h - v_{ij}^h, \qquad \forall (i,j) \in \mathcal{A}.$$

The lower bound of p_i^h is then given by $\max_{j:(i,j)\in\mathcal{A}}\{p_j^h - c_{ij} - u_{ij}^h - v_{ij}^h\}$. If the set of outgoing arcs from node i is empty, we set p_i^h to p_l^h.

Updating u_{ij}^h and v_{ij}^h. For a candidate station $i \in \tilde{\mathcal{N}}$, once p_i^h is decreased, we can potentially decrease the values of u_{ji}^h and v_{ji}^h associated with an incoming arc $(j,i) \in \tilde{\mathcal{A}}$, provided that $u_{ji}^l + v_{ji}^l$ is positive. For feasibility, the new value of $u_{ji}^h + v_{ji}^h$ must be at least $\delta_{ji} = \max(0, p_i^h - p_j^h - c_{ji})$. We then set the values of u_{ji}^h and v_{ji}^h according to their initial proportion, that is,

$$u_{ji}^h = \frac{u_{ji}^l}{u_{ji}^l + v_{ji}^l}\delta_{ji} \quad \text{and} \quad v_{ji}^h = \frac{v_{ji}^l}{u_{ji}^l + v_{ji}^l}\delta_{ji}.$$

This consequently ensures that u_{ji}^h and v_{ji}^h are nonincreasing, and by Proposition 2, the optimality is preserved as the algorithm proceeds.

Node Selection. Given the constraint $p_j^h - p_i^h \leq c_{ij} + u_{ij}^h + v_{ij}^h$ for each arc $(i,j) \in \tilde{\mathcal{A}}$, decreasing p_j^h is beneficial for decreasing u_{ij}^h and v_{ij}^h only if p_j^h is sufficiently larger than p_i^h. This suggests we consider reducing p_i^h with large values first. Therefore, we examine nodes in descending order of p_i^l. Note that the order is determined once at the beginning and used throughout the process.

The complexity of the algorithm is established in the following proposition.

Proposition 3. *For a given origin $l \in \mathcal{L}$, the computational complexity of the basic algorithm is $O(n \log n + k(m + \tilde{m}))$, where $n = |\mathcal{N}|, m = |\mathcal{A}|, \tilde{m} = |\tilde{\mathcal{A}}|$, and $k = |\mathcal{H}^l|$.*

Proof. Sorting p^l requires $O(n \log n)$ time. For each O-D pair in $\mathcal{H}^l$, updating p_i^h's involves scanning outgoing arcs at each node and requires a total of $O(m)$ operations. For each arc $(i,j) \in \tilde{\mathcal{N}}$, u_{ij}^h and v_{ij}^h are updated once when node j is examined. Hence, updating u^h and v^h requires $O(\tilde{m})$ operations. It follows that the overall complexity of the basic algorithm is $O(n \log n + k(m + \tilde{m}))$. □

To demonstrate the algorithm, we provide a small example in Figure 4-1. A label on each arc denotes an arc cost. Note that node 0 is the only origin in this example. There are six integer feasible solutions to this problem: $\{0,4\}$, $\{0,5\}$, $\{0,6\}$, $\{4,5\}$, $\{4,6\}$, $\{5,6\}$, each represents the set of open stations.

For feasible solution $\bar{y} = \{0,6\}$ (that is, $(\bar{y}_0, \bar{y}_4, \bar{y}_5, \bar{y}_6) = (1,0,0,1)$), an optimal solution (p^l, u^l, v^l) to the tree formulation (with an origin $l = 0$) and the set of (p^h, u^h, v^h)'s obtained from the basic algorithm are provided in Table 4.2. Let z denote the total cost in the Benders RMP. The cut generated from (p^l, u^l, v^l) is given by

$$\begin{aligned} z &\geq \sum_{i=1,2,3} (p^l_i - p^l_0) - W^l \left((u^l_{04} + u^l_{05} + u^l_{06}) y_0 + v^l_{04} y_4 + v^l_{05} y_5 + v^l_{06} y_6 \right) \\ &= 10 - 3y_4 - 3y_5. \end{aligned}$$

Notice that for $\bar{y} = \{0,6\}$, this cut yields a lower bound of 10, which is the optimal routing cost when arc $(0,6)$ is available.

Now we will go through the algorithm step-by-step to obtain (p^h, u^h, v^h) for O-D pair $h = (0,3)$.

- Initialize p^h_i to p^l_i for all $i \in \mathcal{N}$, and examine nodes in descending order of p^l_i.

- At node 2, set p^h_2 to its lower bound, which is $\max\{p^h_1 - 2, p^h_3 - 2\} = 1$.

- At node 5, set p^h_5 to its lower bound, which is $p^h_2 - 1 = 0$. The lower bound of $u^h_{05} + v^h_{05}$ is $\max(0, p^h_5 - p^h_0 - c_{05}) = 0$. Decrease v^h_{05} to 0.

- Skip node 3 as it is the destination of this O-D pair.

- At node 1, set p^h_1 to its lower bound, which is $p^h_2 - 2 = -1$.

- For node 6, p^h_6 already attains its lower bound, which is $p^h_3 - 1 = 2$.

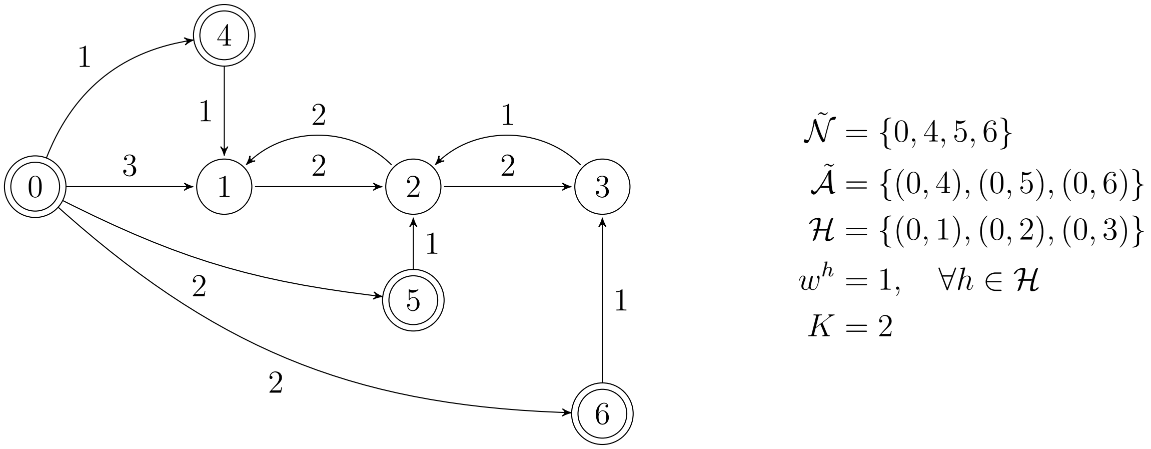

0
1
2
3
4
5
6
1
3
1
2
2
2
1
2
1
2
1
$\tilde{\mathcal{N}} = \{0, 4, 5, 6\}$
$\tilde{\mathcal{A}} = \{(0,4), (0,5), (0,6)\}$
$\mathcal{H} = \{(0,1), (0,2), (0,3)\}$
$w^h = 1, \quad \forall h \in \mathcal{H}$
$K = 2$

Figure 4-1: An example network with four candidate stations and three O-D pairs.

- At node 4, set p_4^h to its lower bound, which is $p_1^h - 1 = -2$. The lower bound of $u_{04}^h + v_{04}^h$ is $\max(0, p_4^h - p_0^h - c_{04}) = 0$. Decrease v_{04}^h to 0.

- The algorithm terminates at node 0.

Note that decreasing v_{04}^h and v_{05}^h to zero corresponds to the fact that making arcs $(0,4)$ or $(0,5)$ available does not provide additional savings to the current optimal path for O-D pair $(0,3)$, which is through node 6.

Similarly, for O-D pair $h = (0,1)$, we can decrease v_{05}^h to zero; and for O-D pair $h = (0,2)$, we can decrease v_{04}^h to zero. Consequently, the (aggregate) cut obtained from our algorithm is given by

$$\begin{aligned} z &\geq \sum_{h \in \mathcal{H}^l} \left[(p_{d(h)}^h - p_{o(h)}^h) - \left((u_{04}^h + u_{05}^h + u_{06}^h) y_0 + v_{04}^h y_4 + v_{05}^h y_5 + v_{06}^h y_6 \right) \right] \\ &= 10 - y_4 - y_5, \end{aligned}$$

which dominates the cut generated from (p^l, u^l, v^l). Additionally, note that for $\bar{y} = \{0, 6\}$, the new cut yields the same lower bound of 10, which indicates that the set of (p^h, u^h, v^h)'s obtained from our algorithm are optimal solutions to the corresponding O-D-based DS as proved in Proposition 2.

So far we have not specified the order in which O-D pairs in $\mathcal{H}^l$ should be processed. Because the algorithm examines nodes in descending order of p_i^l, for O-D pairs h and $h' \in \mathcal{H}^l$ where $p_{d(h)}^l < p_{d(h')}^l$, we have that p_i^h is equal to $p_i^{h'}$ for any node i that is

superscript	p_0	p_4	p_6	p_1	p_3	p_5	p_2	(u_{04}, v_{04})	(u_{05}, v_{05})	(u_{06}, v_{06})
$l = 0$	0	2	2	3	3	3	4	(0,1)	(0,1)	(0,0)
$h = (0, 1)$	0	2	-1	3	0	0	1	(0,1)	(0,0)	(0,0)
$h = (0, 2)$	0	1	2	2	3	3	4	(0,0)	(0,1)	(0,0)
$h = (0, 3)$	0	-2	2	-1	3	0	1	(0,0)	(0,0)	(0,0)

Table 4.2: Optimal solutions to the DS for feasible solution $\bar{y} = \{0, 6\}$

examined before $d(h')$ (i.e., $p^l_i > p^l_{d(h')}$). For example, from Table 4.2, the values of p^h_2 and p^h_5 are the same for O-D pairs $(0, 1)$ and $(0, 3)$. Therefore, we could have avoided repeating the same calculation. To implement this, we process O-D pairs in $\mathcal{H}^l$ in descending order of $p^l_{d(h)}$. Additionally, instead of producing (p^h, u^h, v^h) directly from (p^l, u^l, v^l), we maintain a partially processed solution $(\hat{p}, \hat{u}, \hat{v})$, which is valid for any O-D pair $h \in \mathcal{H}^l$ whose $p^l_{d(h)}$ is less than that of the most recent node examined.

The basic algorithm with this minor improvement is summarized in Algorithm 2. The algorithm begins with initializing a partially processed solution $(\hat{p}, \hat{u}, \hat{v})$ to an optimal solution (p^l, u^l, v^l) to the tree formulation. It then computes a list S of node indices sorted in ascending order of p^l_i (i.e., $S[1] = \text{argmin}_{i \in \mathcal{N}} p^l_i$). Set T denotes a set of O-D pairs in $\mathcal{H}^l$ that have been processed.

The remaining part of the code consists of two nested while-loops. The major difference between the two loops is that the outer loop performs update operations on $(\hat{p}, \hat{u}, \hat{v})$, while the inner loop performs update operations on (p^h, u^h, v^h) for a specific O-D pair $h \in \mathcal{H}^l$. Variable $n(n')$ denotes the number of candidate station nodes that have been examined in the outer(inner) loop; and variable $pos(pos')$ indicates the position of the node currently examined in the outer(inner) loop.

The outer loop repeats until all O-D pairs in $\mathcal{H}^l$ are processed or all candidate station nodes are examined. In the latter case, it means we already considered decreasing the values of $\hat{u}_{ij}$ and $\hat{v}_{ij}$ for every $(i, j) \in \tilde{\mathcal{A}}$, and for each remaining O-D pair $h \in \mathcal{H}^l \setminus T$, we set (p^h, u^h, v^h) to $(\hat{p}, \hat{u}, \hat{v})$ (on line 35).

In the outer loop, the algorithm examines nodes in descending order of p^l_i. At each iteration, if the current node i is a destination of O-D pair $(l, i) \in \mathcal{H}^l$, the algorithm proceeds to compute (p^h, u^h, v^h). Otherwise, $(\hat{p}, \hat{u}, \hat{v})$ is updated accordingly (on lines

superscript	p_0	p_4	p_1	p_5	p_2	p_6	p_3	(u_{04}, v_{04})	(u_{05}, v_{05})	(u_{06}, v_{06})
$l = 0$	0	2	3	4	5	6	7	(1,0)	(0,2)	(0,4)
From the basic algorithm:										
$h = (0, 1)$	0	2	3	1	2	3	4	(1,0)	(0,0)	(0,1)
$h = (0, 2)$	0	2	3	4	5	3	4	(1,0)	(0,2)	(0,1)
$h = (0, 3)$	0	2	3	4	5	6	7	(1,0)	(0,2)	(0,4)
From the improved algorithm (1 pass):										
$h = (0, 1)$	0	2	3	0	1	0	1	(1,0)	(0,0)	(0,0)
$h = (0, 2)$	0	2	3	4	5	3	4	(1,0)	(0,2)	(0,1)
$h = (0, 3)$	0	2	3	4	5	6	7	(1,0)	(0,2)	(0,4)

Table 4.3: Optimal solutions to the DS for feasible solution $\bar{y} = \{0, 4\}$

26-32).

For O-D pair $h = (l, i)$, the algorithm initializes (p^h, u^h, v^h) to $(\hat{p}, \hat{u}, \hat{v})$. Because p_i^h must be fixed, and we no longer need to examine nodes that have already been processed in the outer loop, the inner loop begins with $pos' = pos - 1$ and $n' = n$. The algorithm goes through the remaining nodes and updates (p^h, u^h, v^h) accordingly.

Improved Algorithm

To motivate the idea of the improved algorithm, we consider an optimal solution (p^l, u^l, v^l) to the tree formulation and the set of (p^h, u^h, v^h)'s obtained from the basic algorithm for another feasible solution $\bar{y} = \{0, 4\}$ in Table 4.3. Again, the cut generated from the basic algorithm, $z \geq 15 - 3y_0 - 4y_5 - 6y_6$, dominates the tree cut, $z \geq 15 - 3y_0 - 6y_5 - 12y_6$. Note that for O-D pair $h = (0, 1)$, v_{06}^h is positive, even though arc $(0, 6)$ does not provide any savings. This suggests that we might be able to strengthen the cut further.

The limitation of the basic algorithm is that, for a given O-D pair $h \in \mathcal{H}^l$, each node i is examined for decreasing p_i^h exactly once. We however might be able to decrease p_i^h further after node j that is limiting the lower bound of p_i^h (i.e., $p_i^h = p_j^h - c_{ij} - u_{ij}^h - v_{ij}^h$) is considered. From our example in Table 4.3, for O-D pair $h = (0, 1)$, the lower bound of p_3^h becomes 1 after p_2^h is decreased to 2. If we had decreased p_3^h to 1, then we could decrease p_6^h to 0, and thus reduce v_{06}^h to 0. The

Algorithm 2 The Basic Algorithm

```
Input: $(p^l, u^l, v^l)$
Output: $(p^h, u^h, v^h),\ \forall h \in \mathcal{H}^l$
$(\hat{p}, \hat{u}, \hat{v}) \leftarrow (p^l, u^l, v^l)$
$S \leftarrow$ list of node indices sorted in ascending order of $p^l_i$
$T \leftarrow \emptyset$
$n \leftarrow 0$
$pos \leftarrow |\mathcal{N}|$
while $T \neq \mathcal{H}^l$ and $n < |\tilde{\mathcal{N}}|$ do
    $i \leftarrow S[pos]$
    if $(l, i) = h \in \mathcal{H}^l$ then
        $(p^h, u^h, v^h) \leftarrow (\hat{p}, \hat{u}, \hat{v})$
        $n' \leftarrow n$
        $pos' \leftarrow pos - 1$
        while $n' < |\tilde{\mathcal{N}}|$ do
            $k \leftarrow S[pos']$
            $p^h_k \leftarrow \max_{j:(k,j)\in\mathcal{A}}\{p^h_j - c_{kj} - u^h_{kj} - v^h_{kj}\}$ if $\{j : (k, j) \in \mathcal{A}\} \neq \emptyset$;
                    $p^h_l$ otherwise
            if $k \in \tilde{\mathcal{N}}$ then
                $\delta_{jk} \leftarrow \max(0, p^h_k - p^h_j - c_{jk}),\ \forall (j, k) \in \tilde{\mathcal{A}}$
                $u^h_{jk} \leftarrow \delta_{jk} u^l_{jk}/(u^l_{jk} + v^l_{jk})$ if $(u^l_{jk} + v^l_{jk}) > 0,\ \forall (j, k) \in \tilde{\mathcal{A}}$
                $v^h_{jk} \leftarrow \delta_{jk} v^l_{jk}/(u^l_{jk} + v^l_{jk})$ if $(u^l_{jk} + v^l_{jk}) > 0,\ \forall (j, k) \in \tilde{\mathcal{A}}$
                $n' \leftarrow n' + 1$
            end if
            $pos' \leftarrow pos' - 1$
        end while
        $T \leftarrow T \cup \{h\}$
    end if
    $\hat{p}_i \leftarrow \max_{j:(i,j)\in\mathcal{A}}\{\hat{p}_j - c_{ij} - \hat{u}_{ij} - \hat{v}_{ij}\}$ if $\{j : (i, j) \in \mathcal{A}\} \neq \emptyset$; $\hat{p}_l$ otherwise
    if $i \in \tilde{\mathcal{N}}$ then
        $\delta_{ji} \leftarrow \max(0, \hat{p}_i - \hat{p}_j - c_{ji}),\ \forall (j, i) \in \tilde{\mathcal{A}}$
        $\hat{u}_{ji} \leftarrow \delta_{ji} u^l_{ji}/(u^l_{ji} + v^l_{ji})$ if $(u^l_{ji} + v^l_{ji}) > 0,\ \forall (j, i) \in \tilde{\mathcal{A}}$
        $\hat{v}_{ji} \leftarrow \delta_{ji} v^l_{ji}/(u^l_{ji} + v^l_{ji})$ if $(u^l_{ji} + v^l_{ji}) > 0,\ \forall (j, i) \in \tilde{\mathcal{A}}$
        $n \leftarrow n + 1$
    end if
    $pos \leftarrow pos - 1$
end while
$(p^h, u^h, v^h) \leftarrow (\hat{p}, \hat{u}, \hat{v}),\ \forall h \in \mathcal{H}^l \setminus T$
```

resulting cut is given by $z \geq 15 - 3y_0 - 4y_5 - 5y_6$, which dominates the cut obtained from the basic algorithm.

Therefore, in the improved algorithm, we examine nodes in multiple passes, allowing the changes in nodes with small values of p_i^l to propagate to nodes with higher values of p_i^l. Depending on the size and structure of an underlying network, many passes might be required until no further changes could be made to (p^h, u^h, v^h). To balance the trade-off between solution quality and computational effort, we parameterize the algorithm with a maximum number of passes allowed ($pass$).

The improved algorithm is summarized in Algorithm 3. For brevity, we introduce function Update() in Algorithm 4. Its arguments are a solution (p, u, v) on which the update operations are performed; a starting position, *begin*; an ending position, *end* < *start*; and an O-D pair h under consideration. The function examines nodes in descending order of p_i^h from *begin* to *end* and updates (p, u, v) as usual.

The outer loop is similar to that of the basic algorithm. On line 8, the algorithm reexamines nodes that have already been processed in the outer loop again before computing (p^h, u^h, v^h) for a given O-D pair $h \in \mathcal{H}^l$. Note that the update operations are performed on $(\hat{p}, \hat{u}, \hat{v})$, which is valid for any O-D pair that has not been processed. The Update() call on line 10 is equivalent to the inner loop in the basic algorithm.

At this point, for a given O-D pair $h \in \mathcal{H}^l$, the algorithm has made one pass through all nodes in addition to the updates carried out in the outer loop. On lines 11-17, additional passes through the nodes are performed until there is no change to (p^h, u^h, v^h) or the pass limit is reached.

4.5 Computational Results

In this section, we present computational results that demonstrate the effectiveness of the proposed algorithms in solving the VSND. All algorithms are implemented in Java 1.6 with IBM ILOG CPLEX 12.5 library. Computations are carried out on

Algorithm 3 The Improved Algorithm

```
Input: $(p^l, u^l, v^l), pass$
Output: $(p^h, u^h, v^h),\ \forall h \in \mathcal{H}^l$
$(\hat{p}, \hat{u}, \hat{v}) \leftarrow (p^l, u^l, v^l)$
$S \leftarrow$ list of node indices sorted in ascending order of $p^l_i$
$T \leftarrow \emptyset$
$pos \leftarrow |\mathcal{N}|$
while $T \neq \mathcal{H}^l$ do
    $i \leftarrow S[pos]$
    if $(l, i) = h \in \mathcal{H}^l$ then
        Update$((\hat{p}, \hat{u}, \hat{v}), |\mathcal{N}|, pos + 1, h)$
        $(p^h, u^h, v^h) \leftarrow (\hat{p}, \hat{u}, \hat{v})$
        Update$((p^h, u^h, v^h), pos - 1, 1, h)$
        while $pass > 1$ do
            Update$((p^h, u^h, v^h), |\mathcal{N}|, 1, h)$
            if no change to $(p^h, u^h, v^h)$ then
                break
            end if
            $pass \leftarrow pass - 1$
        end while
        $T \leftarrow T \cup \{h\}$
    end if
    $\hat{p}_i \leftarrow \max_{j:(i,j)\in\mathcal{A}}\{\hat{p}_j - c_{ij} - \hat{u}_{ij} - \hat{v}_{ij}\}$ if $\{j : (i, j) \in \mathcal{A}\} \neq \emptyset$; $\hat{p}_l$ otherwise
    if $i \in \tilde{\mathcal{N}}$ then
        $\delta_{ji} \leftarrow \max(0, \hat{p}_i - \hat{p}_j - c_{ji}),\ \forall (j, i) \in \tilde{\mathcal{A}}$
        $\hat{u}_{ji} \leftarrow \delta_{ji} u^l_{ji} / (u^l_{ji} + v^l_{ji})$ if $(u^l_{ji} + v^l_{ji}) > 0,\ \forall (j, i) \in \tilde{\mathcal{A}}$
        $\hat{v}_{ji} \leftarrow \delta_{ji} v^l_{ji} / (u^l_{ji} + v^l_{ji})$ if $(u^l_{ji} + v^l_{ji}) > 0,\ \forall (j, i) \in \tilde{\mathcal{A}}$
    end if
    $pos \leftarrow pos - 1$
end while
```

Algorithm 4 UPDATE()

```
function UPDATE($(p, u, v), begin, end, h$)
    $pos' \leftarrow begin$
    while $pos' \geq end$ do
        $k \leftarrow S[pos']$
        if $k \neq o(h)$ and $k \neq d(h)$ then
            $p_k \leftarrow \max_{j:(k,j)\in\mathcal{A}}\{p_j - c_{kj} - u_{kj} - v_{kj}\}$ if $\{j : (k,j) \in \mathcal{A}\} \neq \emptyset$;
                $p_{o(h)}$ otherwise
            if $k \in \tilde{\mathcal{N}}$ then
                $\delta_{jk} \leftarrow \max(0, p_k - p_j - c_{jk}),\ \forall (j,k) \in \tilde{\mathcal{A}}$
                $u_{jk} \leftarrow \delta_{jk} u^l_{jk}/(u^l_{jk} + v^l_{jk})$ if $(u^l_{jk} + v^l_{jk}) > 0,\ \forall (j,k) \in \tilde{\mathcal{A}}$
                $v_{jk} \leftarrow \delta_{jk} v^l_{jk}/(u^l_{jk} + v^l_{jk})$ if $(u^l_{jk} + v^l_{jk}) > 0,\ \forall (j,k) \in \tilde{\mathcal{A}}$
            end if
        end if
        $pos' \leftarrow pos' - 1$
    end while
end function
```

Amazon Elastic Compute Cloud (EC2) with 2.5 EC2 Compute Units[1] and 7 GB of RAM.

4.5.1 Instances

The test instances are constructed based on the public transportation network in the Boston metropolitan area. The existing Hubway bike-sharing stations are used as candidate stations. Solving the VSND on these instances essentially determines, if Hubway were to operate with fewer stations—perhaps to limit operating costs, which subset of stations should be installed to minimize overall travel time over the integrated public transportation and VS network. Because bike-sharing service like Hubway is generally used as first- or last-mile transportation, for travel demand, we consider trips originating from each candidate station to the other transit nodes in the existing public transportation network, over 8,000 of them, and presume that the reverse trips would equally benefit from the integrated service.

Additionally, we use the transfer tree concept introduced in Section 3.5.2 to limit

[1] One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

Instance	$\vert\mathcal{N}\vert$	$\vert\tilde{\mathcal{N}}\vert = \vert\mathcal{L}\vert$	$\vert\mathcal{A}\vert$	$\vert\tilde{\mathcal{A}}\vert$	$\vert\mathcal{H}\vert$	K
1	439	8	990	54	1,435	4
2	527	13	1,488	151	2,279	6
3	571	18	1,938	294	3,128	9
4	603	23	2,351	484	4,021	11
5	630	27	2,763	675	4,698	13

Table 4.4: Instances

problem sizes. In particular, at each demand origin, we compute the shortest paths from the origin to the other transit nodes over the existing transit network and derive the corresponding transfer tree. Recall that in a transfer tree, each node represents a transfer point (or an access point if it is adjacent to the origin), and each arc represents the shortest path between the arc's end nodes. The nodes and arcs constituting the transfer trees are then used to construct a graph for each instance. The VS arcs connecting candidate stations are added to the graph, and travel demand to individual transit nodes in the original network is aggregated based on their final transit points. Consequently, each instance is made up of overlaying transfer trees originated from every candidate station. When a candidate station is adjacent to nodes that are also in the transfer trees originating from other candidate stations, the VS arcs linking the stations enable potential multi-modal trips.

The details of the instances are summarized in Table 4.4. For each instance, we consider the number of open stations equal to $K = \lfloor |\tilde{\mathcal{N}}|/2 \rfloor$, which results in the largest (or second largest if $|\tilde{\mathcal{N}}|$ is odd) number of feasible solutions $\binom{|\tilde{\mathcal{N}}|}{K}$. Note that the number of O-D pairs considered in each case is quite large relative to the number of candidate stations.

4.5.2 Problem Formulation and Pareto Optimal Cuts

In this section, we present a set of computational results to illustrate the impacts of problem formulation—multi-cut formulation and tree formulation—and Pareto optimal cuts. In particular, we consider two ways of formulating the VSND: (1) the O-D formulation (presented in Section 4.3.2) and (2) the tree formulation (presented in

Section 4.4.1). At each iteration, in addition to solving the DS, we may also solve the AP to obtain a PO cut. Lastly, because of the separability of the DS, different numbers of cuts can be added to the RMP. We consider adding (1) $|\mathcal{H}|$ cuts, one for each O-D pair, (2) $|\mathcal{L}|$ cuts, one for each origin, and (3) a single aggregate cut. The results obtained using the smallest instance are summarized in Table 4.5.

Consistently with our discussion earlier, the O-D formulation yields stronger Benders cuts than the tree formulation, as indicated by the numbers of iterations. The tree formulation, however, can significantly reduce the computation times. In this case, with the tree formulation, we need to solve only 8 optimization problems, one for each origin, to obtain a usual cut at each iteration; while with the O-D formulation, we need to solve 1,435 optimization problems. Nevertheless, for larger instances, the slow convergence resulting from tree cuts can potentially worsen the overall computation time. We will demonstrate this in the section. This comparison between the O-D and tree formulation substantiates the potential advantages of our solution approach, which allows us to obtain an optimal solution to the O-D-based DS quickly from an optimal solution to the tree-based DS.

The results show that adding multiple cuts at each iteration are effective in improving the convergence of the Benders algorithm. In fact, with a single tree cut added to the RMP at each iteration, the algorithm ends up enumerating all $\binom{8}{4} = 70$ feasible solutions. Given the large number of O-D pairs considered in each instance, adding $|\mathcal{H}|$ cuts to the RMP at each iteration quickly makes it more difficult to solve as the algorithm proceeds and potentially increases the overall computation time as discussed in de Camargo et al. (2008). For this reason, the $|\mathcal{L}|$-cut formulation is used to obtain the computational results presented in the next section.

Finally, PO cuts are proven to perform better than usual cuts in most cases for both O-D and tree formulations. In particular, the results suggest that the computation time savings from the reduced number of iterations outweigh the additional time required for solving the AP.

		O-D based		Tree based	
		Pareto	Usual	Pareto	Usual
$\lvert\mathcal{H}\rvert$-cut	# iterations	5	5	N/A	N/A
	# cuts	7175	7175		
	runtime (secs)	16.74	16.13		
$\lvert\mathcal{L}\rvert$-cut	# iterations	5	7	16	28
	# cuts	40	56	128	224
	runtime (secs)	14.82	23.25	1.32	1.89
1-cut	# iterations	7	16	70	70
	# cuts	7	16	70	70
	runtime (secs)	23.81	51.24	4.45	4.29

Table 4.5: Computational results demonstrating the impacts of problem formulation and Pareto optimal cuts.

4.5.3 Algorithm Performance

In this section, we compare the performance of different algorithms for solving the VSND. To obtain the results in this section, we implement a slightly modified version of the classical Benders decomposition algorithm (Algorithm 1). Specifically, we limit the RMP solving time to the amount of time spent generating cuts in the previous iteration or until a feasible integral solution is found. This allows us to balance the time spent generating new cuts and solving the RMP and iterate the Benders Algorithm more quickly. In the last iteration, however, the RMP is still solved to optimality to ensure the optimality of the incumbent solution at termination. Note that this change only affects later iterations at which the RMP becomes too difficult to solve to optimality within the time limit. The adverse effect of early termination is that some unnecessary cuts generated from suboptimal solutions to the RMP are added to the formulation. Nonetheless, in many cases, the MIP solver can easily find the optimal solution to the RMP within the time limit, but more computation time is required only to verify its optimality.

Tables 4.6 and 4.7 summarize the computational results for the Benders decomposition approach with 11 types of cuts and the traditional branch-and-cut approach using CPLEX. The 11 types of cuts are PO cuts from both O-D and tree formulations, usual cuts from the O-D formulation, cuts obtained from the basic algorithm (Algorithm 2), and cuts obtained from the improved algorithm (Algorithm 3) with the maximum number of passes ranging from 1 to 45. In the basic and improved algorithms, we use an optimal solution to the tree-based AP as an initial solution. Lastly, in the branch-and-cut approach, the tree formulation is used as the O-D formulation can only be solved for the smallest instance due to memory limit. The solving time is limited to two hours.

Comparing the first two columns in Table 4.6, we conclude that PO cuts are indeed effective in accelerating the Benders decomposition algorithm. The results for PO cuts generated from the O-D and tree formulations confirm that a tight formulation yields stronger Benders cuts, as indicated by the numbers of iterations.

Our proposed algorithms are proven to be effective in strengthening Benders cuts. Even with the basic algorithm, which involves minimal computational effort, the resulting cuts require significantly smaller numbers of iterations to converge compared to the PO tree cuts, which are used as initial solutions in the algorithm. More importantly, in terms of overall computation time, the cuts generated from the basic algorithm outperform the PO O-D cuts in the first four instances, despite the larger numbers of iterations.

Additionally, the results clearly demonstrate that the improved algorithm can effectively strengthen Benders cuts further. As the pass limit increases, the number of iterations required becomes smaller. Remarkably, in many cases, the improved algorithm results in smaller numbers of iterations than that of the PO O-D cuts. Overall computation time, however, does not necessarily decrease as we increase the pass limit. In particular, the computation time savings resulting from fewer iterations might not make up for the extra computation time for strengthening the cuts. From the results, we conclude that to achieve the best computation time, the maximum number of passes should be increased, that is strengthening the cuts more aggressively, as the problem size grows.

The advantages of the Benders decomposition algorithm are more pronounced in the larger instances. Especially for Instance 5, the Benders algorithm takes less than 13 minutes to solve to optimality; while the branch-and-cut approach fails to obtain a lower bound to ensure optimality of the incumbent solution, which is in fact optimal, within the time limit of two hours. As discussed in Section 4.4, this is partly because the tree formulation used in the branch-and-cut approach is not tight, and the optimal cost of its LP relaxation does not provide a good lower bound to the original mixed integer program. In conclusion, this suggests that the Benders decomposition algorithm together with our cut generation algorithm is a more effective exact method for solving the VSND.

It is important to note that in addition to its effectiveness, the Benders decomposition approach offers reoptimization capabilities (Geoffrion and Graves, 1974; Cordeau et al., 2006), which are particularly useful for strategic planning, as in this context. Specifically, when we modify the problem such that it does not affect the feasible region of the Benders dual subproblem, the extreme points found when solving the original problem are still valid and can be used to generate Benders cuts for the new problem. In fact, if the changes only affect the master problem (e.g., adding side constraints), the Benders cuts previously generated can be used without additional computation. This allows us to avoid solving the modified problem from scratch and obtain a new optimal solution in fewer iterations, while for the branch-and-bound approach, solving the modified problem from scratch is inevitable. In our context, the reoptimization capabilities enable us to reexamine the problem with different numbers of stations K, travel demand estimates w^h, travel costs c_{ij}, and/or additional logical constraints on the network configuration (for example, at least three stations must be installed in each partnered municipality).

4.6 Conclusions and Future Work

This chapter addresses passenger-centric strategic planning for one-way vehicle-sharing systems. We develop optimization models for locating vehicle-sharing stations to minimize overall travel time over the integrated vehicle-sharing and public transportation network. Benders decomposition is used to tackle large instances. We present and

Instances		Pareto O-D	Usual O-D	Pareto Tree	Basic Algorithm	Improved Algorithm ($pass = 1$)
1	# iterations	5	7	16	9	8
	optimal soln time (secs)	11.77	13.24	0.60	0.79	1.44
	runtime (secs)	14.82	23.25	1.32	1.69	2.09
2	# iterations	7	16	42	16	14
	optimal soln time (secs)	57.54	91.71	11.82	5.59	6.52
	runtime (secs)	57.61	119.11	14.49	6.58	7.59
3	# iterations	18	63	418	105	108
	optimal soln time (secs)	155.19	53.94	290.00	38.34	10.53
	runtime (secs)	277.63	1283.74	1034.59	205.55	254.16
4	# iterations	40	93	761	362	151
	optimal soln time (secs)	911.98	-	-	716.92	441.83
	runtime (secs)	1842.76	7200.00	7200.00	1830.32	959.16
	optimality gap	-	0.15%	0.34%	-	-
5	# iterations	55	70	590	659	619
	optimal soln time (secs)	3985.57	-	-	-	-
	runtime (secs)	7200.00	7200.00	7200.00	7200.00	7200.00
	optimality gap	-	0.47%	0.16%	0.22%	0.16%

Table 4.6: Computational results comparing performance of different algorithms (1)

Instances		Improved Algorithm						CPLEX
		pass = 2	*pass* = 3	*pass* = 5	*pass* = 10	*pass* = 15	*pass* = 45	
1	# iterations	7	6	6	6	6	6	-
	optimal soln time (secs)	1.64	2.30	2.62	4.79	8.48	21.05	0.20
	runtime (secs)	2.84	3.54	4.01	7.20	10.30	25.41	1.43
2	# iterations	12	9	7	7	7	7	-
	optimal soln time (secs)	9.29	9.37	10.44	16.67	24.36	63.08	3.22
	runtime (secs)	9.36	10.47	10.50	19.58	28.30	73.76	4.54
3	# iterations	59	51	29	19	12	11	-
	optimal soln time (secs)	42.88	46.06	25.49	73.40	42.52	159.07	24.40
	runtime (secs)	115.32	118.24	91.53	93.52	106.31	278.32	37.59
4	# iterations	118	64	33	25	23	21	-
	optimal soln time (secs)	371.93	305.35	168.35	179.23	241.02	478.66	147.32
	runtime (secs)	829.11	482.40	220.20	226.66	302.52	741.84	826.13
	optimality gap	-	-	-	-	-	-	-
5	# iterations	295	134	64	43	28	25	-
	optimal soln time (secs)	2718.74	1986.49	702.66	540.84	649.66	1314.14	1001.83
	runtime (secs)	5165.69	2510.24	1207.37	859.29	754.69	1576.71	7200.00
	optimality gap	-	-	-	-	-	-	-

Table 4.7: Computational results comparing performance of different algorithms (2)

discuss two classes of formulations: the O-D and tree formulations. While an aggregate formulation—the tree formulation in this case—is generally discouraged in the literature as it yields weaker Benders cut, and hence slower convergence, we exploit it to generate Benders cuts quickly and propose novel algorithms to strengthen the resulting cuts further. Using data from the Boston metropolitan area, we present computational results that confirm the effectiveness of our solution approach.

There are three particularly interesting extensions of this work for future research.

- *Application to other network design problems.* The idea of obtaining Benders cuts quickly using aggregate formulations, as presented in this work, is applicable to any network design problems whose Benders subproblem involves solving a network flow problem, and whose commodities or O-D pairs have a common origin (or destination) that can be grouped together. Cut strengthening algorithms, however, need to be tailored for different problems.

- *Integrating with local search heuristics.* While this work is focused primarily on developing an exact solution method for the problem, the capability to obtain good solutions quickly might also be helpful, in the initial phases of the planning process, for getting insights into good network designs for the city and/or understanding the effects of different parameters and constraints. Therefore, it would be interesting to integrate local search heuristics with our exact solution approach to improve the incumbent solutions at early iterations. Rei et al. (2009), among others, have demonstrated the successful use of the local branching technique of Fischetti and Lodi (2003) to accelerate Benders decomposition.

- *Incorporating operational considerations into the model.* In order to keep the optimization models tractable, we do not explicitly consider vehicle movements in this work and assume that commuters can always find a vehicle and a docking slot available at any station. Nevertheless, careful tactical and operational planning with appropriate decisions on station capacity, fleet size, and fleet relocation operations can help limit the extent to which the assumption is violated.

Chapter 5

Concluding Remarks

Public transportation is undeniably an effective way, economically and environmentally, to move a large number of people in a city. It reduces traffic congestion, fuel consumption, and carbon footprint. Its ineffectiveness, such as long travel times, poor coverage, and lack of direct services, however, makes it unappealing to many commuters. In this thesis, we present analytics and optimization approaches to improving public transit and its integration with vehicle-sharing services.

In Chapter 2, we propose to improve existing bus services through introduction of limited-stop bus services, which have the advantage of shorter in-vehicle travel times for passengers. Focusing on incremental changes to the existing schedule, we seek to reassign some number of local bus trips to operate a limited-stop service. We present an optimization model to determine optimal limited-stop service route and frequency that maximize total user welfare. In this model, we introduce a novel way to capture user behavioral changes in response to a new limited-stop service. Theoretical properties of the model are established and used to develop an efficient solution approach. The use of data visualization in this process (see Figures 2-5, 2-8, 2-9) is proven to be helpful for understanding the model and structure of optimal solutions. Using data from a bus operator in a major city, we present numerical results and discussions on solution quality, computational times, and sensitivity of parameters. In addition to the methodological contributions, we provide insights into key characteristics of a bus service that affect the potential benefits of limited-stop services.

In Chapters 3 and 4, we propose to improve urban mobility through better integration between existing public transportation and one-way vehicle-sharing services. A proper design of one-way vehicle-sharing networks can provide better access to existing public transportation and additional options for trips beyond those provided by public transit, especially in the areas with low travel demand, where it is economically infeasible to provide high levels of traditional public transportation services.

We present, in Chapter 3, a framework for evaluating the impacts of integrated vehicle-sharing and public transportation services. We describe our approach to modeling public transit graphs representing integrated multi-modal transportation services, which allows us to assess the impacts of vehicle-sharing networks at the level of O-D's. Based on centrality indices, we propose accessibility, utilization, and efficiency metrics for measuring the impacts from different aspects. To facilitate understanding of changes in commuting patterns resulting from vehicle-sharing services, we introduce the notion of a transfer tree and develop a novel approach to visualize a transfer hierarchy of trips originating from a given transit node. Through a case study of Boston, we demonstrate the use of evaluation metrics in conjunction with the visualization tool. Because the framework utilizes publicly available data and web services, transportation engineers and urban planners can apply it to evaluate a potential or existing vehicle sharing network in any city whose transit schedules are published in the increasingly adopted GTFS format. Additionally, we believe that the visualization tool can help enhance communication between different stakeholders in the planning process. In fact, a slightly different version of our visualization tool[1] was selected as a winner of the Hubway Data Visualization Challenge[2] by a panel of judges consisting of representatives from Hubway, the Metropolitan Area Planning Council, the School of Architecture and Planning at MIT, and the Boston Globe.

Given the benefits of integrating vehicle-sharing and public transit services, we present, in Chapter 4, a methodology for passenger-centric strategic planning for one-

[1] http://hubway.virot.me

[2] http://hubwaydatachallenge.org/

way vehicle-sharing systems. We present optimization models for selecting vehicle-sharing station locations to minimize overall travel time over the integrated network. Because the problem consists of two sets of decisions that are made sequentially—a vehicle-sharing network design and optimal commuting paths, it is natural to use Benders decomposition as a solution approach to tackle large instances. We discuss two classes of formulations: the O-D and tree formulations. While an aggregate formulation—the tree formulation in this case—is generally discouraged in the literature as it yields weaker Benders cut, and hence slower convergence, we exploit it to generate Benders cuts quickly and propose novel algorithms to strengthen the resulting cuts further. Using data from the Boston metropolitan area, we present computational results that confirm the effectiveness of our solution approach. In addition to its effectiveness, the reoptimization capabilities of Benders decomposition make it particularly more appealing for strategic planning, as in this context.

Bibliography for Chapter 1

American Public Transportation Association. Transit Savings Report, August, 2012. URL http://www.publictransportation.org/tools/transitsavings/Pages/default.aspx.

T. Hodges. *Public Transportation's Role in Responding to Climate Change*. 2010.

T. Lomax, D. Schrank, and S. Turner. Annual urban mobility report. *Texas Transportation Institute*, 2011.

United Nations Population Fund. State of World Population 2007: Unleashing the Potential of Urban Growth. Technical report, 2007.

U.S. Census Bureau. American Community Survey 2010. URL http://www.census.gov/acs/www/data_documentation/2010_release/.

Bibliography for Chapter 2

A. Ceder. Optimal design of transit short-turn trips. *Transportation Research Record*, 1989.

A. Ceder. *Public transit planning and operation : theory, modelling and practice*. Elsevier, London; Burlington, MA, 2007. ISBN 9780750661669 0750661666.

A. Ceder and H. Stern. Deficit Function Bus Scheduling with Deadheading Trip Insertions for Fleet Size Reduction. *Transportation Science*, 15(4):338–363, Nov. 1981. ISSN 0041-1655. doi: 10.1287/trsc.15.4.338.

A. Ceder and N. Wilson. Bus network design. *Transportation Research Part B: Methodological*, 20(4):331–344, Aug. 1986. ISSN 01912615. doi: 10.1016/0191-2615(86)90047-0.

G. Desaulniers and M. Hickman. Public Transit. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, chapter 2, pages 69–127. Elsevier, 2007. doi: 10.1016/S0927-0507(06)14002-5.

P. Furth. Alternating Deadheading in Bus Route Operations. *Transportation Science*, 19(1):13–28, Feb. 1985. ISSN 0041-1655. doi: 10.1287/trsc.19.1.13.

P. Furth. Zonal Route Design for Transit Corridors. *Transportation Science*, 20(1):1–12, Feb. 1986. ISSN 0041-1655. doi: 10.1287/trsc.20.1.1.

P. Furth. Short Turning on Transit Routes. *Transportation Research Record*, 1987.

V. Guihaire and J. Hao. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251–1273, Dec. 2008. ISSN 09658564. doi: 10.1016/j.tra.2008.03.011.

W. Jordan and M. Turnquist. Zone Scheduling of Bus Routes to Improve Service Reliability. *Transportation Science*, 13(3):242–268, Aug. 1979. ISSN 0041-1655. doi: 10.1287/trsc.13.3.242.

H. Larrain, R. Giesen, and J. Muñoz. Choosing the Right Express Services for Bus Corridor with Capacity Restrictions. *Transportation Research Record: Journal of the Transportation Research Board*, 2197:63–70, Dec. 2010. ISSN 0361-1981. doi: 10.3141/2197-08.

C. Leiva, J. Muñoz, R. Giesen, and H. Larrain. Design of limited-stop services for an urban bus corridor with capacity constraints. *Transportation Research Part B: Methodological*, 44(10):1186–1201, 2010. ISSN 0191-2615. doi: 10.1016/j.trb.2010.01.003.

S. Schwarcz. Service design for heavy demand corridors: limited-stop bus service. Master thesis, Massachusetts Institute of Technology, 2004.

H. Scorcia. Design and evaluation of BRT and limited-stop services. Master thesis, Massachusetts Institute of Technology, 2010.

P. Tétreault and A. El-Geneidy. Estimating bus run times for new limited-stop service using archived AVL and APC data. *Transportation Research Part A: Policy and Practice*, 44(6):390–402, July 2010. ISSN 09658564. doi: 10.1016/j.tra.2010.03.009.

Transportation Research Board. Estimating the Benefits and Costs of Public Transit Projects: A Guidebook for Practitioners. *TCRP Report*, 78, 2002.

Transportation Research Board. Bus Rapid Transit Practitioner's Guide. *TCRP Report*, 118, 2007.

Y. Ulusoy, S. Chien, and C. Wei. Optimal All-Stop, Short-Turn, and Express Transit Services Under Heterogeneous Demand. *Transportation Research Record: Journal of the Transportation Research Board*, 2197(-1):8–18, Dec. 2010. ISSN 0361-1981. doi: 10.3141/2197-02.

Bibliography for Chapter 3

R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1 edition, 1993. ISBN 013617549X.

H. Bast, E. Carlsson, A. Eigenwillig, R. Geisberger, C. Harrelson, V. Raychev, and F. Viger. Fast routing in very large public transportation networks using transfer patterns. In M. de Berg and U. Meyer, editors, *Algorithms ESA 2010*, volume 6346 of *Lecture Notes in Computer Science*, pages 290–301. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-15774-5. doi: 10.1007/978-3-642-15775-2_25.

M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, 1985.

P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J.-B. Rouquier, and E. Fleury. Shared bicycles in a city: A signal processing and data analysis perspective. *Advances in Complex Systems*, 14(03):415–438, 2011. doi: 10.1142/S0219525911002950.

Central Transportation Planning Staff. Transfer penalties in urban mode choice modeling. Technical report, 1997. prepared for the Travel Model Improvement Program, FTA, U.S. Department of Transportation.

R. Cervero, A. Golub, and B. Nee. City CarShare: Longer-Term Travel Demand and Car Ownership Impacts. *Transportation Research Record: Journal of the Transportation Research Board*, 1992:70–80, Jan. 2007. doi: 10.3141/1992-09.

P. Crucitti, V. Latora, and S. Porta. Centrality in networks of urban streets. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 16(1):015113, 2006. doi: 10.1063/1.2150162.

C. Curtis and J. Scheurer. Planning for sustainable accessibility: Developing tools to aid discussion and decision-making. *Progress in Planning*, 74(2):53 – 106, 2010. ISSN 0305-9006. doi: 10.1016/j.progress.2010.05.001.

P. DeMaio. Bike-sharing: history, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):41–56, 2009.

S. Derrible. Network centrality of metro systems. *PLoS ONE*, 7(7):e40575, 07 2012. doi: 10.1371/journal.pone.0040575.

S. Derrible and C. Kennedy. Network analysis of world subway systems using updated graph theory. *Transportation Research Record: Journal of the Transportation Research Board*, 2112:17–25, 2009.

S. Derrible and C. Kennedy. The complexity and robustness of metro networks. *Physica A: Statistical Mechanics and its Applications*, 389(17):3678 – 3691, 2010. ISSN 0378-4371. doi: 10.1016/j.physa.2010.04.008.

E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. ISSN 0029-599X. 10.1007/BF01386390.

L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

L. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215 – 239, 1978–1979. ISSN 0378-8733. doi: 10.1016/0378-8733(78)90021-7.

Z. Guo and N. Wilson. Assessment of the transfer penalty for transit trips geographic information system-based disaggregate modeling approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1872:10–18, 2004.

Z. Guo and N. Wilson. Modeling effects of transit system transfers on travel behavior: Case of commuter rail and subway in downtown boston, massachusetts. *Transportation Research Record: Journal of the Transportation Research Board*, 2006:11–20, 2007.

Z. Guo and N. Wilson. Assessing the cost of transfer inconvenience in public transport systems: A case study of the london underground. *Transportation Research Part A: Policy and Practice*, 45(2):91 – 104, 2011. ISSN 0965-8564. doi: 10.1016/j.tra.2010.11.002.

T. Hodges. *Public Transportation's Role in Responding to Climate Change*. 2010.

R. Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003. ISSN 1530-2415. doi: 10.1111/j.1530-2415.2003.00015.x.

C. Lane. PhillyCarShare: First-Year Social and Mobility Impacts of Carsharing in Philadelphia, Pennsylvania. *Transportation Research Record: Journal of the Transportation Research Board*, 1927:158–166, Jan. 2005. doi: 10.3141/1927-18.

V. Latora and M. Marchiori. Efficient Behavior of Small-World Networks. *Physical Review Letters*, 87, 2001. doi: 10.1103/PhysRevLett.87.198701.

V. Latora and M. Marchiori. Is the boston subway a small-world network? *Physica A: Statistical Mechanics and its Applications*, 314(1-4):109 – 113, 2002. ISSN 0378-4371. doi: 10.1016/S0378-4371(02)01089-0.

E. Martin and S. Shaheen. Greenhouse gas emission impacts of carsharing in north america. *Intelligent Transportation Systems, IEEE Transactions on*, 12:1074–1086, 2011.

E. Martin, S. Shaheen, and J. Lidicker. Impact of carsharing on household vehicle holdings. *Transportation Research Record: Journal of the Transportation Research Board*, 2143:150–158, 2010.

MBTA. *Ridership and Service Statistics*. 13 edition, 2010.

M. Müller-Hannemann, F. Schulz, D. Wagner, and C. Zaroliagis. Timetable Information: Models and Algorithms. volume 4359 of *Lecture Notes in Computer Science*, pages 67–90. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-74245-6. doi: 10.1007/978-3-540-74247-0_3.

R. Nair, E. Miller-Hooks, R. Hampshire, and A. Bui. Large-scale vehicle sharing systems: Analysis of vlib'. *International Journal of Sustainable Transportation*, 7 (1):85–106, 2013. doi: 10.1080/15568318.2012.660115.

S. Porta, P. Crucitti, and V. Latora. The network analysis of urban streets: a primal approach. *Environment and Planning B: Planning and Design*, 33(5):705–725, 2006.

E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. Efficient models for timetable information in public transportation systems. *ACM Journal on Experimental Algorithmics*, 12:2.4:1—-2.4:39, June 2008. ISSN 1084-6654. doi: 10.1145/1227161.1227166.

S. Raveau, J. C. Muñoz, and L. de Grange. A topological route choice model for metro. *Transportation Research Part A: Policy and Practice*, 45(2):138 – 147, 2011. ISSN 0965-8564. doi: 10.1016/j.tra.2010.12.004.

C. Rydén and E. Morin. Mobility services for urban sustainability. environmental assessment. *Report WP 6. Trivector Traffic AB*, 2005.

J. Scheurer and S. Porta. Centrality and connectivity in public transport networks and their significance for transport sustainability in cities. *World Planning Schools Congress, Mexico, July 13-16, 2006*, 2006.

S. Shaheen and A. Cohen. Carsharing and personal vehicle services: Worldwide market developments and emerging trends. *International Journal of Sustainable Transportation*, 7(1):5–34, 2012.

S. Shaheen, S. Guzman, and H. Zhang. Bikesharing across the Globe. In J. Pucher and R. Buehler, editors, *City Cycling*, chapter 9. The MIT Press, 2012a.

S. Shaheen, E. Martin, A. Cohen, and R. Finson. Public bikesharing in north america: Early operator and user understanding. Technical report, San Jose, CA, June 2012b.

J. Sienkiewicz and J. Hołyst. Statistical analysis of 22 public transport networks in poland. *Physical Review E*, 72(4):046127, 2005.

H. Spiess and M. Florian. Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological*, 23(2):83 – 102, 1989. ISSN 0191-2615. doi: 10.1016/0191-2615(89)90034-9.

J. Stasko, R. Catrambone, M. Guzdial, and K. Mcdonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 53(5):663 – 694, 2000. ISSN 1071-5819. doi: 10.1006/ijhc.2000.0420.

C. von Ferber, T. Holovatch, Y. Holovatch, and V. Palchykov. Public transport networks: empirical analysis and modeling. *The European Physical Journal B - Condensed Matter and Complex Systems*, 68:261–275, 2009. ISSN 1434-6028. doi: 10.1140/epjb/e2009-00090-x.

I. Vragović, E. Louis, and A. Díaz-Guilera. Efficiency of informational transfer in regular and complex networks. *Physical Review E*, 71:036122, Mar 2005. doi: 10.1103/PhysRevE.71.036122.

Bibliography for Chapter 4

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. GERAD Tech. Rep. G-2012-57, HEC Montréal, Canada, 2012.

A. Awasthi, D. Breuil, S. Chauhan, M. Parent, and T. Reveillere. A multicriteria decision making approach for carsharing stations selection. *Journal of Decision Systems*, 16(1):57–78, 2007. doi: 10.3166/jds.16.57-78.

A. Awasthi, S. Chauhan, X. Hurteau, and D. Breuil. An Analytical Hierarchical Process-based decision-making approach for selecting car-sharing stations in medium size agglomerations. *International Journal of Information and Decision Sciences*, 1(1):66–97, Jan. 2008. doi: 10.1504/IJIDS.2008.020049.

M. Benchimol, P. Benchimol, B. Chappert, A. de la Taille, F. Laroche, F. Meunier, and L. Robinet. Balancing the stations of a self service bike hire system. *RAIRO - Operations Research*, 45:37–61, 0 2011. ISSN 1290-3868. doi: 10.1051/ro/2011102.

J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

J. Birge and F. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384 – 392, 1988. ISSN 0377-2217. doi: 10.1016/0377-2217(88)90159-2.

D. Chemla, F. Meunier, and R. Calvo. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, 10(2):120 – 146, 2013. ISSN 1572-5286. doi: 10.1016/j.disopt.2012.11.005.

C. Contardo, C. Morency, and L.-M. Rousseau. Balancing a dynamic public bike-sharing system. Working paper, CIRRELT, 2012.

I. Contreras and E. Fernández. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680 – 697, 2012. ISSN 0377-2217. doi: 10.1016/j.ejor.2011.11.009.

I. Contreras, J.-F. Cordeau, and G. Laporte. Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 59(6):1477–1490, 2011. doi: 10.1287/opre.1110.0965.

I. Contreras, J.-F. Cordeau, and G. Laporte. Exact solution of large-scale hub location problems with multiple capacity levels. *Transportation Science*, 46(4):439–459, 2012. doi: 10.1287/trsc.1110.0398.

J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35 (4):375–388, 2001. doi: 10.1287/trsc.35.4.375.10432.

J.-F. Cordeau, F. Pasin, and M. Solomon. An integrated model for logistics network design. *Annals of Operations Research*, 144(1):59–82, 2006. ISSN 0254-5330. doi: 10.1007/s10479-006-0001-3.

A. M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429 – 1450, 2005. ISSN 0305-0548. doi: 10.1016/j.cor.2003.11.012.

G. Côté and M. Laughton. Large-scale mixed integer programming: Benders-type heuristics. *European Journal of Operational Research*, 16(3):327 – 333, 1984. ISSN 0377-2217. doi: 10.1016/0377-2217(84)90287-X.

M. Daskin. *Network and discrete location: models, algorithms, and applications.* Wiley-Interscience, 2011.

R. de Camargo, G. Miranda Jr., and H. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research*, 35(4):1047 – 1064, 2008. ISSN 0305-0548. doi: 10.1016/j.cor.2006.07.002.

M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1-3): 23–47, 2003. ISSN 0025-5610. doi: 10.1007/s10107-003-0395-5.

B. Fortz and M. Poss. An improved benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359 – 364, 2009. ISSN 0167-6377. doi: 10.1016/j.orl.2009.05.007.

A. Geoffrion and G. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20(5):822–844, 1974. doi: 10.1287/mnsc.20.5.822.

H. Hamacher and Z. Drezner. *Facility location: applications and theory.* Springer Verlag, 2002.

A. Kek, R. Cheu, Q. Meng, and C. Fung. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):149–158, 2009.

D. Kim, C. Barnhart, K. Ware, and G. Reinhardt. Multimodal express package delivery: A service network design application. *Transportation Science*, 33(4):391–407, 1999. doi: 10.1287/trsc.33.4.391.

P. Kumar and M. Bierlaire. Optimizing locations for a vehicle sharing system. In *Proceedings of the Swiss Transport Research Conference*, Ascona, Switzerland, 2012.

J. Lin and T. Yang. Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E: Logistics and Transportation Review*, 47(2):284–294, Mar. 2011. ISSN 13665545. doi: 10.1016/j.tre.2010.09.004.

T. Magnanti and R. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981. doi: 10.1287/opre.29.3.464.

T. Magnanti and R. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984. doi: 10.1287/trsc.18.1.1.

T. Magnanti, P. Mireault, and R. Wong. Tailoring benders decomposition for uncapacitated network design. In G. Gallo and C. Sandi, editors, *Netflow at Pisa*, volume 26 of *Mathematical Programming Studies*, pages 112–154. Springer Berlin Heidelberg, 1986. ISBN 978-3-642-00922-8. doi: 10.1007/BFb0121090.

D. McDaniel and M. Devine. A modified benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319, 1977. doi: 10.1287/mnsc.24.3.312.

A. Mercier, J.-F. Cordeau, and F. Soumis. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451 – 1476, 2005. ISSN 0305-0548. doi: 10.1016/j.cor.2003.11.013.

M. Minoux. Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360, 1989. ISSN 1097-0037. doi: 10.1002/net.3230190305.

P. Mirchandani and R. Francis. *Discrete location theory.* John Wiley & Sons, New York, 1990.

N. Mladenović, J. Brimberg, P. Hansen, and J. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

R. Nair. *Design and Analysis of Vehicle Sharing Programs: A Systems Approach.* PhD thesis, University of Maryland, 2010.

R. Nair and E. Miller-Hooks. Fleet management for vehicle sharing operations. *Transportation Science*, 2010. doi: 10.1287/trsc.1100.0347.

J. Naoum-Sawaya and S. Elhedhli. An interior-point benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, pages 1–23, 2010. ISSN 0254-5330. doi: 10.1007/s10479-010-0806-y.

G. Nemhauser and L. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

N. Papadakos. Integrated airline scheduling. *Computers & Operations Research*, 36 (1):176 – 195, 2009. ISSN 0305-0548. doi: 10.1016/j.cor.2007.08.002.

T. Raviv and O. Kolka. Optimal inventory management of a bike-sharing station. *IIE Transactions*, 2013. doi: 10.1080/0740817X.2013.770186. To appear.

T. Raviv, M. Tzur, and I. Forma. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, pages 1–43, 2013. ISSN 2192-4376. doi: 10.1007/s13676-012-0017-6.

J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006. ISSN 1097-0037. doi: 10.1002/net.20128.

W. Rei, J.-F. Cordeau, M. Gendreau, and P. Soriano. Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345, 2009. doi: 10.1287/ijoc.1080.0296.

J. Shu, M. Chou, Q. Liu, C.-P. Teo, and I.-L. Wang. Bicycle-sharing system: deployment, utilization and the value of re-distribution. Working paper, National University of Singapore, 2010.

H. Smith, G. Laporte, and P. Harper. Locational analysis: highlights of growth to maturity. *Journal of the Operational Research Society*, 60:S140–S148, Feb. 2009. ISSN 0160-5682. doi: 10.1057/jors.2008.172.

H. Spiess and M. Florian. Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological*, 23(2):83 – 102, 1989. ISSN 0191-2615. doi: 10.1016/0191-2615(89)90034-9.

P. Tsamasphyrou, A. Renaud, and P. Carpentier. Transmission network planning under uncertainty with benders decomposition. In V. Nguyen, J.-J. Strodiot, and P. Tossings, editors, *Optimization*, volume 481 of *Lecture Notes in Economics and Mathematical Systems*, pages 457–472. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-66905-0. doi: 10.1007/978-3-642-57014-8_30.