Strategic network planning for emerging mobility services

March 2022

Kai Zhang

Strategic network planning for emerging mobility services

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2022

Kai Zhang

Abstract

In recent years, new mobility services have flourished, offering more user-centric transportation products. With the emergence of these services, the transport systems are experiencing significant changes. How to design these emerging systems from a long-term perspective becomes a challenging issue. This thesis focuses on strategic planning for two specific types of emerging mobility services: bus transit systems operating with human-driven and autonomous buses and one-way station-based carsharing systems. By using mathematical formulations and approaches, various strategic decisions are determined in the two respective systems.

For designing the emerging bus transit systems, a mixed integer nonlinear programming (MINLP) model (Model I) and its linear approximation are proposed to decide bus depot locations, fleet size, bus routes, and frequency and bus type of each route under the restriction of bus drivers. Considering the complexity of linearized Model I, we develop a simplified path-based model (Model II) and its linear approximation to solve the joint design problem. For the approximately linear models, we conduct the experiments on two test networks. Based on different weights, we first handle the linearized Model I and obtain the efficient frontiers for the networks. The results reveal the importance of autonomous buses in constructing user-centric service systems when the driver is limited. On both networks, the linearized Model II can be solved faster than the linearized Model I due to the smaller model size. To evaluate the solutions of the linearized Model II, we compare the objective values of original Model I and new Model I that includes constraints on the solutions of model II. It turns out that the linearized Model II can obtain satisfactory solutions for both networks within a few seconds.

For the one-way station-based carsharing systems, this thesis focuses on strategic decisions, including the location and capacity of stations and the fleet size. Under demand uncertainty, we introduce a two-stage risk-averse stochastic model to maximize the mean return and minimize the risk, where conditional value-at-risk (CVaR) is specified as the risk measure. To solve the problem efficiently, we develop a branch-and-cut algorithm and a scenario decomposition algorithm. We generate the scenario demand data with the Poisson distribution based on limited historical use data to conduct computational experiments. In the experimental part, the efficient frontiers are obtained firstly so that the system operator can make a trade-off between return and risk. We then utilize an evaluation method to analyze the necessity of introducing risk. Finally, the efficiency of the proposed algorithms is elaborated through comparative experiments. Both the branch-and-cut algorithm and the scenario decomposition algorithm can tackle the small- and medium-scale problems well. For large-scale problems that cannot be solved by using an optimization solver or the branch-and-cut algorithm, the scenario decomposition method can provide favorable solutions within a reasonable time.

Keywords: Strategic planning, Mathematical formulation, Autonomous buses, Path-based, Carsharing, Risk, Demand uncertainty, Branch-and-cut, Scenario decomposition

Acknowledgements

Time flies, and the three-year doctoral study at the University of Tsukuba is coming to an end. The knowledge, harvesting, and life experiences I have learned in these three years have undoubtedly been precious treasures in my life. On the occasion of completing my thesis, I would like to sincerely thank those who have given me help, care, and support.

First of all, I want to sincerely thank my supervisor, Prof. Akiko Yoshise, for her patient and meticulous guidance. Without her selfless help, I could not finish this thesis. Prof. Yoshise has deeply influenced me in the past three years with her rigorous academic attitude. When I felt confused, she was always willing to give me useful advice.

Then, I would like to express my gratitude to Prof. Yuichi Takano and Prof. Maiko Shigeno for their valuable suggestions. Special thanks also to the other members of my thesis committee: Prof. Yoshiaki Osawa and Prof. Yu Song, for their insightful suggestions and comments. Based on their advice, I can improve this work greatly.

Besides, I want to thank Toyota Motor Corporation for providing the data on the Ha:mo RIDE carsharing system. I am grateful to the China Scholarship Council for providing financial support during my Ph.D. study. I would also like to thank my colleague, Yuzhu Wang, for his kind help in my study and life.

Last but not least, I would like to thank my parents in particular. Living overseas is sometimes lonely, and I cannot hold on without their encouragement and support. It is their support that makes me fearless in the face of difficulties.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Along with urbanization and economic growth, the continuous increase in private vehicles has had severe negative impacts, such as traffic congestion, environmental pollution, time wastage, and shortage of parking spaces. On the other hand, these negative impacts become the critical factors promoting the wave of innovation in transportation [21, 85]. Transportation agencies have to reasonably design and operate the transport systems by employing emerging technologies to offer more efficient, flexible, and sustainable services. In recent years, the concept of mobility has gained growing popularity and restructured traditional transportation gradually. The mobility services aim to provide more user-centric transportation products with a high degree of flexibility to respond to travelers' needs, habits, and preferences. [85] introduced some notable types of emerging mobility services (e.g., carsharing, ridesharing, shared autonomous vehicles, and mobility-as-a-service (MaaS)). Figure 1.1 [85] presents the ideal travel range and the use flexibility of different mobility services. Each service is suitable for a particular travel situation, and users can choose the appropriate mode for the trips. Compared with traditional transportation, the states of emerging mobility are now mainly transforming in two aspects: from human-driven to autonomous and from private to shared [23]. This thesis studies two related emerging mobility services, bus transit with mixed human-driven and autonomous buses and one-way station-based carsharing, for these two types of transformations.

In the past decade, autonomous vehicles have rapidly advanced thanks to the devel-

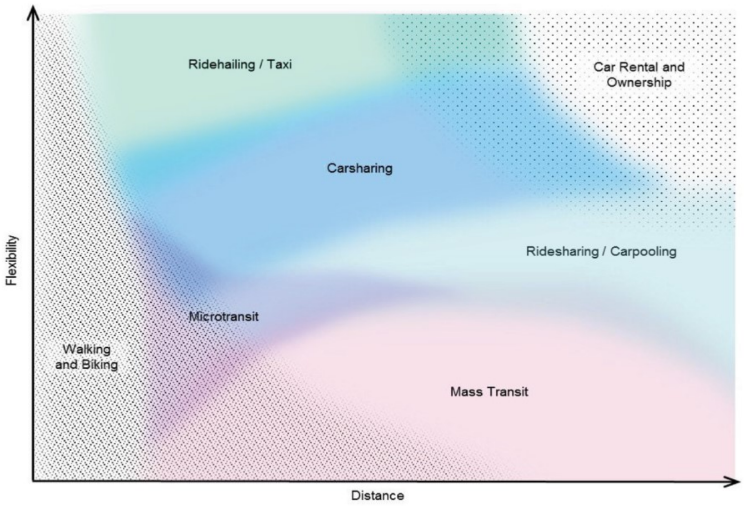


Figure 1.1: Ideal use cases for different mobility services [85]

opment of sensors, wireless connectivity, and artificial intelligence. According to the J3016 standard for automated driving systems by the Society of Automotive Engineers (SAE), vehicle autonomy can be classified into six levels, ranging from no driving automation (Level 0) to full driving automation (Level 5) [81]. Litman [60] predicted that, in the 2050s, about half of vehicles would be autonomous if Level 5 vehicles were commercially available in the 2030s. Due to the vast market, many traditional car companies (such as Toyota, Honda, and General Motors) have researched and developed autonomous vehicle prototypes, even some technological giants (Apple, Google, and Tesla). The developed autonomous vehicles are expected to be used for different purposes like mobility, logistics, and product sales. Among various vehicle types, buses and trucks may be automated the fastest because of high labor costs and lack of drivers [60, 96]. Transit buses, a kind of mass transit, play a crucial role in modern cities and bring many benefits, such as alleviating congestion and reducing the dependency on private vehicles [32]. However, the shortage of professional bus drivers results in bus operators canceling certain specific routes or reducing departure frequency. The introduction of autonomous buses seems to relieve this condition. At the end of 2020 and the beginning of 2021, some countries (e.g., Japan [71], Singapore [98], America [8], and China [88]) have successively launched the fixed-route autonomous bus trials. As some countries seek to conduct pilot tests on autonomous buses, designing efficient bus systems operating with a mixed human-driven and autonomous fleet becomes essential.

Carsharing is another emerging mobility service to promote sustainable transport besides the change in public transit systems. Traditional multimodal public transportation can satisfy most travel demands, but many people prefer private vehicles due to their greater convenience, especially those who reside far away from public transportation stations. In Figure 1.1, it is observed that private cars are more suitable for long-distance travel and can provide higher flexibility than other mobility services. However, the cost of owning a car and parking difficulties bring their own challenges. Recently, carsharing has become alternative transportation mean and can offer the advantages of both private vehicles and public transportation [20,53]. Although carsharing is a new mobility service for many cities and is usually considered innovative, the first carsharing project can date back to 1948 in Zurich, Switzerland [68,99]. The early shared car systems were not successful for economic reasons, and real prosperity appeared in the 2000s because of the more common integration of information collection systems and mobile services [35]. By now, many carsharing organizations have boomed around the world [92]. The existing carsharing systems can be mainly classified into one-way and two-way (round-trip) types [82]. In one-way systems, users can pick up and return cars at different sites, while in two-way systems, users should return rented cars to the site where they were picked up. Compared with two-way systems, one-way systems are more convenient for users, considering that one-way trips usually occupy a large percentage of the total trips [5,26]. Furthermore, station-based and free-floating systems can be distinguished in view of parking-spot restrictions [82]. The former type forces people to park vehicles at stations with limited parking spaces, whereas the latter allows the users to park cars anywhere in an operation area. In recent years, one-way station-based carsharing has been witnessing soaring popularity worldwide. However, the system operators are still facing significant challenges in planning and operating their systems.

1.2 Research scope

With the emergence of heterogeneous fleet bus transit systems and one-way station-based carsharing systems, it becomes important to plan them properly. However, designing and managing such systems tend to be difficult and raise many decision-making problems, including mainly three dimensions: strategic (long-term), tactic (medium-term), and operational (short-term) [47, 50]. Sometimes, even real-time controls are needed because of undesired situation [47]. The strategic decisions provide an overall structure for the systems, which significantly influence the performance of subsequent decisions at other levels. In this thesis, we make strategic plans for both systems by using mathematical programming approaches.

Although the two systems involve different strategic decisions, facility location and fleet size are issues that both systems need to solve. Facility location problems are widespread in the strategic planning procedure. In such problems, facilities are located at candidate sites to minimize the cost of satisfying the demands under some constraints. As revealed in [30], locating facilities is a long-term investment that extensively influences subsequent decisions. This study solves the bus depot (also called garage, center) location problem (BDLP) and carsharing station location problem for bus transit and carsharing systems, respectively.

It is also important to determine the fleet size in each system. Diverse vehicles are necessary for pick-up and delivery purposes. In many distribution-routing problems, a natural question arises as to how many and what types of vehicles are required to satisfy the demand [39]. Zakaria [99] argued that a good choice of fleet size could affect the quality of service (QoS) significantly. Therefore, in our models, we determine the number of buses (human-driven & autonomous) in the bus transit network and the number of shared mobilities in the carsharing system.

In addition to the issues above, the bus transit network design entails three other problems: setting the bus routes and their corresponding frequencies and bus types. The set of routes over the underlying network constitute the transit network. Our objective is to choose a subset of routes from a predefined set of routes. The frequency setting problem is an essential portion of transit network design. The frequencies have a considerable effect on the passenger waiting times and the service quality [32], so they should be properly evaluated. With limited driver resources, we shall consider assigning human-driven and autonomous buses to appropriate routes to minimize costs. These three problems (routes, frequencies, and bus types) are collectively referred to as a transit network design problem (TNDP) in the thesis, though different studies have various definitions.

1.3 Thesis structure

This thesis is organized into five chapters. The remaining chapters are as follows. In Chapter 2, we review the research regarding bus transit systems and carsharing systems, two main focuses of this thesis, and identify current research gaps. Chapter 3 solves a bus depot location and transit network design problem (BDL&TNDP) in the context of mixed human-driven and autonomous buses. A mixed integer nonlinear programming model (MINLP) and its linear approximation are proposed to determine the bus depot locations, fleet size, bus routes, and frequency and bus type of each route. In addition, a simplified path-based model is given to solve the design problem in a heuristic way. Chapter 4 presents a risk-averse two-stochastic MINLP model to optimize the strategic decisions including on locations and capacities of stations and fleet size considering uncertain demand for one-way station-based carsharing. In order to obtain more robust solutions, we consider risk management and apply conditional value-at-risk (CVaR) to measure risk. Two algorithms are developed from different perspectives to solve the problem. Finally, Chapter 5 concludes this thesis and provides some future research directions.

Chapter 2

Literature review

This chapter demonstrates a detailed review of the related studies on bus transit and carsharing systems, especially the papers on applying optimization methods to design them strategically. At the end of this chapter, we summarize the research gaps and clarify the contributions of this thesis.

2.1 Bus transit systems

Planning problems in bus transit systems have been widely studied, spanning from strategic decisions (e.g., infrastructure development) to real-time control policies (e.g., stop-skipping strategies). Figure 2.1 summarizes the planning problems at different levels [18, 47, 78]. The complete planning problem of the system is intractable if approached as a whole. Therefore, it is usually divided into a number of manageable subproblems that are treated sequentially during the planning process. Interested readers can refer to [29] and [47] for more comprehensive reviews of different dimensions. Note that there is no unique definition of each problem [32]. Models with the same names may solve different problems and vice versa, depending on the papers' definitions. In this study, we handle the problem BDL&TNDP for the emerging bus transit systems to determine bus depot locations, bus routes, service frequencies, and bus type of each route.

Regrading the mixed BDL&TNDP, there have been very few studies so far. Sawicki and Fierek [77] handled a BDL&TNDP without frequency setting based on a macro-simulation framework. In the framework, the passenger assignment was simulated by a four-stage

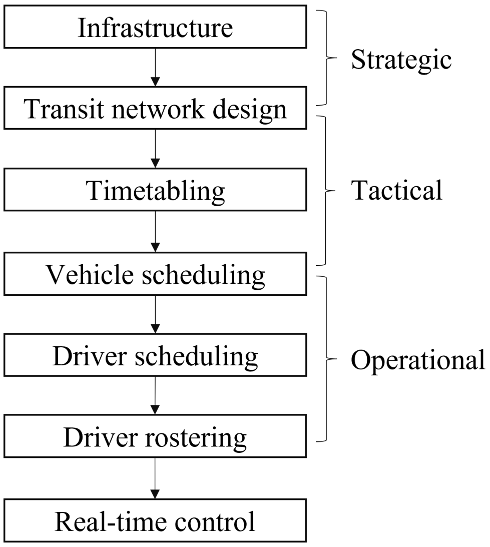


Figure 2.1: Common transit planning problems

procedure, while line construction and depot location were determined in an optimization model. Their methodology was validated on a medium-sized transport network in Poland. Liu et al. [61] simultaneously optimized the transit routes, service frequency, and location of charging depots for an electric bus transit network by formulating a multi-objective optimization model that takes into account constraints on bus route, charging depot, vehicle operation, and charging schedule. Passenger assignment was based on an assignment procedure in [100] without well-defined mathematical constraints. The problem was solved by Pareto artificial fish swarm algorithm.

Although scarce literature paid attention to BDL&TNDP, many studies were conducted on BDLP or TNDP. Next, we also review studies about BDLP and TNDP, considering that BDL&TNDP integrates the two problems.

2.1.1 Bus depot location problem (BDLP)

BDLP is a typical location-allocation problem, which usually determines the optimal location of bus depots and assigns the buses to different routes. In such a location problem, minimizing the total cost of the transit system is a common objective function, and the dead mileage cost accounts for a significant share in total cost [19]. Dead mileage (or dead running, deadheading) means the empty travel distance between the starting/ending bus stops and the depots without serving any passengers. With urban development, bus depots can generally be set up in undeveloped areas far away from the city center due to limited land space [43], which may result in a significant increase in dead mileage cost. However, it can be reduced by reasonably locating the depots and allocating the buses, thereby decreasing the total costs.

Back in the 1980s, Maze et al. [65, 66] studied BDLP to locate and size the bus depots with the traditional discrete location model. The objective function comprised dead mileage, depot operating, and depot construction costs. In [66], an efficient solution method was proposed and applied to a Detroit metropolitan area case study. Uyeno and Willoughby [90] formulated a cost-minimizing mixed integer linear programming (MILP) model to determine the optimal number, location, and size of transit centers based on existing bus routes and transit centers. Some existing centers may be shut down in the model, and some new centers may be opened. Following the study, Willoughby and Uyeno [94] solved a similar problem and developed a two-step heuristic procedure. The heuristic procedure first assigned all buses on a route to the same center and then considered capacity limitations to reallocate. Willoughby [93] applied the location model for a regional transit system in Vancouver, Canada. They also analyzed the model in different transit planning scenarios, such as no candidate facilities, forced Oakridge allocation, and Greenfields approach. Extending the formulation in [93], Ali and Hassan [1] recently integrated the maintenance center availability to the bus depot location problem. The developed model took into account the construction cost of new depots, the residual value of existing depots, the cost of maintenance centers located at depots, and the dead mileage cost. Chen et al. [19] suggested a queuing-location-allocation model to design bus depot systems. Their study treated the bus depot as a congested facility and developed an M/M/S queuing problem [67]. A Lagrangian relaxation algorithm was proposed to solve the problem.

2.1.2 Transit network design problem (TNDP)

TNDP is a complex variant of the network design problem and can be formulated as a mathematical model. However, it is often challenging to model various aspects of the problem [16] and obtain the optimal solution for large-scale instances [64]. Therefore, many studies tackled the problem through heuristics or metaheuristics without explicit mathematical programming formulations [2–4, 33, 69]. Recently, Iliopoulou et al. [49] provided a critical review of numerous papers applying metaheuristics to handle TNDP and presented an exhaustive performance comparison among them. Although heuristics and metaheuristics may be the only practical way to solve large-scale problems, mathematical programming models allow for precise analysis of the relationships between different system components, which provides insight into the problem structure and possible solution methods. [16].

The early TNDP models were broadly classified into cost-oriented and passenger-oriented approaches [78, 79]. The former approaches attempted to find a set of bus routes serving all passengers and minimizing the operators' costs (e.g., [40, 91]), while the latter ones sought the maximal number of direct travelers (e.g., [14]). It can be found that the early models ignored the passengers' behavior (passenger assignment) that can be formulated by the optimization model. Schöbel and Scholl [79] are the pioneers who considered the passengers' transfer behavior of users by constructing a change&go network. The model's objective is to minimize the travel time of all passengers and the number of transfers needed. In 2012, Schöbel [78] summarized the basic line-planning models and some approaches integrating line planning and passenger assignment. Szeto and Jiang [86] proposed a bi-level MINLP model in which the bus routes and corresponding frequencies are determined at the upper level. The passengers' behavior, including waiting for the buses, is modeled by using an optimal strategies assignment model [84] at the lower level. Given the difficulty of solving the bi-level model, a hybrid artificial bee colony algorithm was implemented to settle two real-world transit networks. Later, Cancela et al. [16] linearized the optimal strategies assignment model approximately to tackle TNDP and incorporated a budget constraint representing the operators' interest into the linearized model. They also discussed a bi-level formulation by analyzing transfer behavior and street and bus capacity constraints. Recently, Duran et al. [31] illustrated a pollution-transit network design problem (P-TNDP) and its three different cases to plan a multimodal transit system considering the network congestion and CO_2 emissions. For a particular case: the unimodal transit system, a bi-objective MILP model was proposed, which was an extension of the model in [16]. De-Los-Santos et al. [28] elaborated two exact formulations for bus stop location and line planning problems on the underlying bus-pedestrian network to minimize social welfare. Besides waiting and transfer costs, their models also took the walking time into account to represent the entire travel process of users. The models were tested on a real metropolitan network with Seville city in Spain and nine surrounding towns.

With the piloting of fixed-route autonomous bus projects in several countries, a few researchers began investigating the TNDP operating such new mobilities. Bergqvist and Åstrand [6] studied the effects of automated minibusses on existing transit systems in terms of operating costs and environmental impact. The best combination of traditional buses and autonomous minibuses was derived to minimize the total operating costs. Tian et al. [89] integrated autonomous vehicles into transit systems and formulated a two-stage stochastic MINLP model to optimize the service frequencies of the conventional buses and autonomous buses on the pre-determined routes, thus obtaining the fleet size of each type of bus. Considering the nonconvex nature of their problem, they reformulated it as a mixed-integer quadratic program, which can be solved by the approach of quadratic transform with a linear alternating algorithm. In [42], the authors established a framework composed of a multi-objective optimization problem and a multi-objective artificial bee algorithm to explore potential changes in network design and frequency settings when autonomous buses were introduced to fixed-route bus networks.

2.2 Carsharing systems

Similar to bus transit systems, design and management of a carsharing system also raise several decision-making problems, ranging from strategic issues (e.g., carsharing mode, station location) to operational policies (e.g., vehicle relocation, staff scheduling) [9], [50], as shown in Figure 2.2. Generally, the overall structure of a carsharing system is determined at the strategic and tactical levels, which have a significant impact on system performance. Operational problems are usually handled to improve system management. Regarding optimization problems at different levels, interested readers can refer to [11] and [37] for a comprehensive review. More recently, Illgen and Höck [50] presented a systematic review

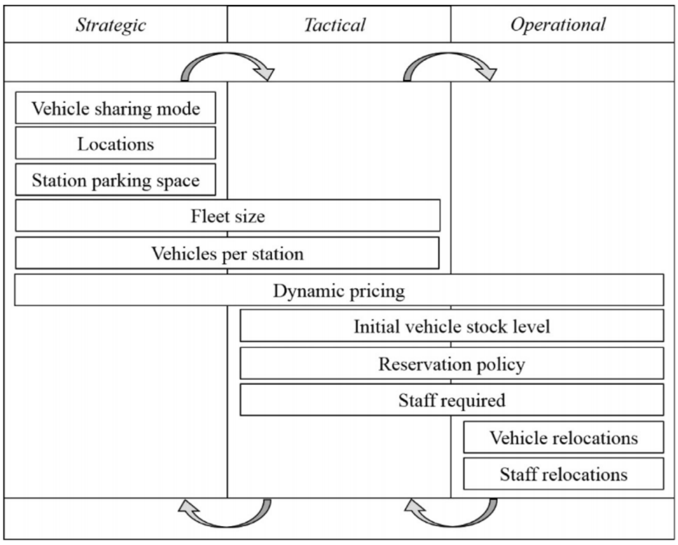


Figure 2.2: Common carsharing planning problems [50]

centered around a key operational issue: the vehicle relocation problem (VReP) in one-way carsharing systems and introduced dependent-decision problems at other levels associated with VReP. Over the past decade, VReP has become the most commonly considered problem [10,13,34,45,52,70,95,102] . For example, Boyaci et al. [10] developed an optimization-simulation framework for both vehicle and personnel relocations in the electric carsharing systems with reservations. Three mathematical models were included in the framework: station clustering, operations optimization, and personnel flow.

Compared with the VReP, only a little literature has concentrated on strategic decisions involving location problems, as revealed by Çalık and Fortz [15]. The classic facility location problems (e.g., covering, median, center problems) have been researched for more than half a century, and there are a large number of related studies. Daskin [24] provided a thorough review in terms of models, algorithms, and applications. Within the context of carsharing, de Almeida Correia and Antunes [25] are pioneers who first solved the station location problem in one-way systems. In their study, three MILP models were proposed and compared to determine the optimal number, location, and size of stations with the same objective that maximizes the profit. By modifying the constraints, they incorporated different schemes (i.e., serve some selected requests, serve all requests, and reject some requests conditionally) into the models. The practicality of the models was illustrated with a case study on Lisbon, Portugal. Following this study, Jorge et al. [51] developed an agent-based simulation model considering demand variability and vehicle relocation policy. Experiments were conducted on the Lisbon data to test the validity of the solution from the MILP model proposed by [25] . Similarly, de Almeida Correia et al. [26] evolved the MILP formulation in [25] to allow users to pick up the sharing vehicles at an alternate station other than the closest one. Later, Boyacı et al. [9] formulated a multi-objective MILP model for planning one-way electric carsharing systems considering vehicle relocation and EV charging requirements. The objectives of the model are to maximize the profit of the operator and the profit of the users. Because of the large number of relocation variables, the problem becomes intractable in real-world cases. The relocation variables were therefore reduced by grouping the demand and clustering the stations to form an aggregate model. The proposed approach was validated with data from Nice, France. Huang et al. [46] showed an MINLP model to fathom the station location and capacity problem with a nonlinear demand rate represented by the logit model. To solve the MINLP model, a customized gradient algorithm was developed. The model and algorithm were applied to Suzhou Industrial Park, Suzhou, China.

Additionally, some recent studies began to take the demand uncertainty into account when making strategic decisions, whereas most of the earlier carsharing research, including demand uncertainty, focused on dealing with VReP. Brandstätter et al. [12] identified the optimal locations and the number of required EVs for one-way station-based carsharing systems by introducing a stochastic optimization model whose objective is to maximize the expected profit. In that model, the uncertain demand is represented by several scenarios. Due to the complexity of the problem, they proposed a heuristic algorithm to obtain an approximate solution or to provide an initial heuristic solution. Lu et al. [62] introduced a stochastic model to optimize the profitability and QoS considering uncertain one-way and two-way rental demand for a hybrid reservation-based and free-floating system. The desired QoS level was maintained by minimizing the expected penalty of unserved customers in a risk-neutral model. To solve the stochastic model, they developed a branch-and-cut algorithm with mixed-integer rounding-enhanced Benders cuts. They also briefly remarked on a risk-averse model by penalizing the CVaR of unserved demand, but did not give corresponding detailed experimental results. Çalık and Fortz [15] modeled the location problem as a two-stage stochastic MILP model and developed a Benders decomposition algorithm to solve it. As in Brandstätter et al. [12], the objective function maximized the expected profit. On the basis of a demand prediction model, many demand scenarios were generated from Manhattan taxi trip data and numerical experiments were conducted. The results illustrated that the proposed algorithm could help reduce computing time and obtain favorable solutions. Zhang et al. [101] presented a risk-averse MINLP to determine station locations, station capacities, and fleet size considering the uncertain demand. A branch-and-cut algorithm and a scenario decomposition algorithm were used to deal with the model. Moreover, they compared their solution methods with the Benders decomposition algorithm in [62] and demonstrated the superiority of their methods.

2.3 Research gaps and our contributions

As for emerging bus transit systems, the BDL&TNDP has not been extensively researched. Although some studies (e.g., [42]) investigated the TNDP for autonomous bus systems, none of them considered BDLP to locate the bus depots and allocate human-driven and autonomous buses to the routes with limited driver resources when designing the network. Existing studies such as [61] and [77] noted BDL&TNDP, but did not take into account the mixed fleet and the explicit mathematical programming formulations on passenger assignment to analyze the users' cost. Besides, most papers directly used meta-heuristics to deal with TNDP or BDL&TNDP after providing a mathematical formulation. None paid attention to the model structure to simplify the model to obtain a solution. Therefore, to fill the research gaps, we contribute to the literature in the following aspects:

- proposing a hybrid model (Model I) that puts together different modeling sides: operators' and users' costs, bus allocation, passenger assignment, a mixed fleet with different travel costs, and driver restriction to solve the BDL&TNDP;
- introducing a simplified path-based formulation (Model II) to approximate the users' costs by assigning passengers to predefined paths freely and guessing the waiting

time heuristically, thus obtaining approximate solutions in a short time. Unlike the general path-based model in [7] without considering waiting time, our model, which is simplified from Model I, estimates waiting time in an approximate way.

Regarding one-way station-based carsharing systems, we can see that there has been little research on determining the location and capacity of stations and the fleet size simultaneously while considering stochastic demand. Some studies like [9, 12, 15] merely took part of them into account and ignored the potential risk. [62] is the only work that involves the risk in carsharing studies to the best of our knowledge. Lu et al. [62] briefly introduced a model that applied CVaR to reduce the unserved demand, but they did not conduct the corresponding experiments. Rare carsharing studies have considered the risk to avoid heavy losses. Compared with the previous research, our main contributions are threefold:

- formulating a new risk-averse stochastic MINLP model to solve the joint design problem, where CVaR is specified as the risk measure to identify tail losses;
- developing two different algorithms: a branch-and-cut algorithm and a scenario decomposition algorithm, to handle the proposed model by converting the original model into two equivalent forms;
- analyzing the model with risk term by using a training and testing method and comparing the computational performance among our solution methods and the improved Benders decomposition algorithm illustrated in [62].

Chapter 3

Joint optimization of bus depot location and transit network design considering a heterogeneous fleet

3.1 Introductory remarks

Bus transit systems have been indispensable in modern cities, helping enhance the mobility of passengers in a sustainable and affordable manner. The development of bus transit is considered a promising solution to mitigate traffic congestion and air pollution caused by the dramatic increase in private vehicles. However, many countries with mature public transit systems have recently faced a shortage of professional bus drivers, leading bus operators to cancel certain specific routes or reduce the departure frequency. Hence, the introduction of autonomous vehicles in transit services seems to be an alternative to compensate for the driver shortage in the era of autonomous driving. Compared to human-driven buses, the operating costs of autonomous buses can reduce by about 50% for no hiring and training costs [42, 59].

This chapter focuses on a BDL&TNDP for the emerging bus transit systems with the operation of human-driven and autonomous buses to determine bus depot locations, fleet size, a set of routes, and the frequency and bus type of each route considering a limited number of drivers. To solve the problem, we formulate an MINLP model (Model I) with the objective of minimizing the weighted sum of operators' and users' costs. In the model, the optimal strategies passenger assignment is taken into account to estimate the users' costs, including in-vehicle travel time and waiting time. To handle the MINLP model, we linearize the model approximately with the method in [16]. It is still challenging to solve the problem despite the linear approximation. The model is formulated over a trajectory network and computes flow volume and waiting time at each node (node-based), which dramatically increases the complexity. Hence, we develop a new simplified path-based model (Model II) and its approximately linear model. The models assign passengers to predefined travel paths directly rather than the arcs and assume an expected waiting time for each route rather than each node. In order to evaluate Model II, we compare the efficiency of solving the two models and analyze the performance of the solutions of Model II by substituting them into Model I. Two networks in previous studies are used to verify the models. Because nonlinear models are complicated, all our experiments and comparisons are based on linearized Models I and II.

The chapter is organized as follows. Section 3.2 describes the problem setting and detailed formulations of Model I. The formulations of Model II are shown in Section 3.3. Section 3.4 reports the experiments and results. Finally, the chapter ends in Section3.5.

3.2 Modle I

3.2.1 Problem setting and assumptions

In order to minimize both operators' and users' costs, the proposed model determines the proper depot location sites, fleet size, bus routes, and the frequency and bus type of each route in transit systems operating with human-driven and autonomous buses. Operators' costs are composed of capital and operating costs in the model, while the total travel time is adopted to represent users' costs.

To formulate the model, we first consider a physical road graph $G^R = (V, E)$, where each vertex $v \in V$ represents a possible bus stop, and each edge $e \in E$ represents a bidirectional street connecting two different nodes. For vertex $j \in J \subseteq V$, the bus depot can be located to store and maintain the bus fleet. We intend to find an optimal set of depot location $J^* \subseteq J$ to reduce the operating cost wasted on dead mileage without serving any passengers. Besides, the optimal route set $R^* \subseteq R$ and the optimal corresponding frequency $f^* \in \mathbb{R}_+$ and bus type $h^* \in H$ for each route $r \in R^*$ are determined as well, where R and H represent the sets of bus routes and bus types, respectively. Given bus type $h \in H$, the travel time on edge $e \in E$ is c_{eh}. We denote the set of edges forming the route $r \in R$ by E_r. Based on the physical road network, the passengers (users) can travel from their origins to their destinations by utilizing the bus routes R^*. The demand for origin-destination (OD) pair $i \in I$ is δ_i whose origin is $O_i \in V$ and destination is $D_i \in V$.

When the users' costs are took into account in the formulation, it becomes vital to model the behavior of passengers so as to identify which route or routes will be rode. The route choices can usually be demonstrated by a transit assignment model of optimal strategies [84]. Such a model is built on a trajectory network defined by the road graph and the given bus routes, see Appendix A. In this study, we apply an extended trajectory graph that considers different bus types. The extended trajectory graph is denoted as $G^T = (N, A)$, which is directed. The nodes in the set N consists of two parts: route nodes (dummy nodes) N^R and physical nodes V, that is, $N = N^R \cup V$. The set N^R includes the generated nodes n^h_{rv} for each route $r \in R$ through the vertex $v \in V$ using the bus type $h \in H$. Moreover, the arcs that connect different nodes are classified into three disjoint categories: waiting arcs A^W, in-vehicle arcs A^I, and destination arcs A^D. $A = A^W \cup A^I \cup A^D$. In detail, for each node $v \in V$, each route $r \in R$ and each bus type $h \in H$, the waiting arc connects the physical node v and the generated route node n^h_{rv}, and the destination arc connects the generated route node n^h_{rv} and the physical node v. Hence, we have $A^W = \bigcup_{v \in V, r \in R, h \in H} \left\{(v, n^h_{rv})\right\}$ and $A^D = \bigcup_{v \in V, r \in R, h \in H} \left\{(n^h_{rv}, v)\right\}$. For OD pair $i \in I$, the waiting arcs including node O_i have zero cost; otherwise, there are penalty costs of other waiting arcs due to the transfer behavior of passengers. The costs of destination arcs are always zero. For route r and bus type h, the edge $e = (v_1, v_2) \in E_r$ can generate a forward arc $(n^h_{rv_1}, n^h_{rv_2})$ and a backward arc $(n^h_{rv_2}, n^h_{rv_1})$ so that $A^I = \bigcup_{r \in R, (v_1, v_2) \in E_r, h \in H} \left\{(n^h_{rv_1}, n^h_{rv_2}), (n^h_{rv_2}, n^h_{rv_1})\right\}$. With respect to the costs of in-vehicle arcs, we have $c_{(n^h_{rv_1}, n^h_{rv_2})} = c_{(n^h_{rv_2}, n^h_{rv_1})} = c_{eh}$, where $e = (v_1, v_2)$. The trajectory graph is directed where the node order of an arc represents the travel direction. To illustrate more clearly, Figure 3.1 gives an example of the extended trajectory graph for a physical road network with three bus routes. Note that we generate in-vehicle arcs for each type of bus, as the in-vehicle travel time varies for different types of buses.

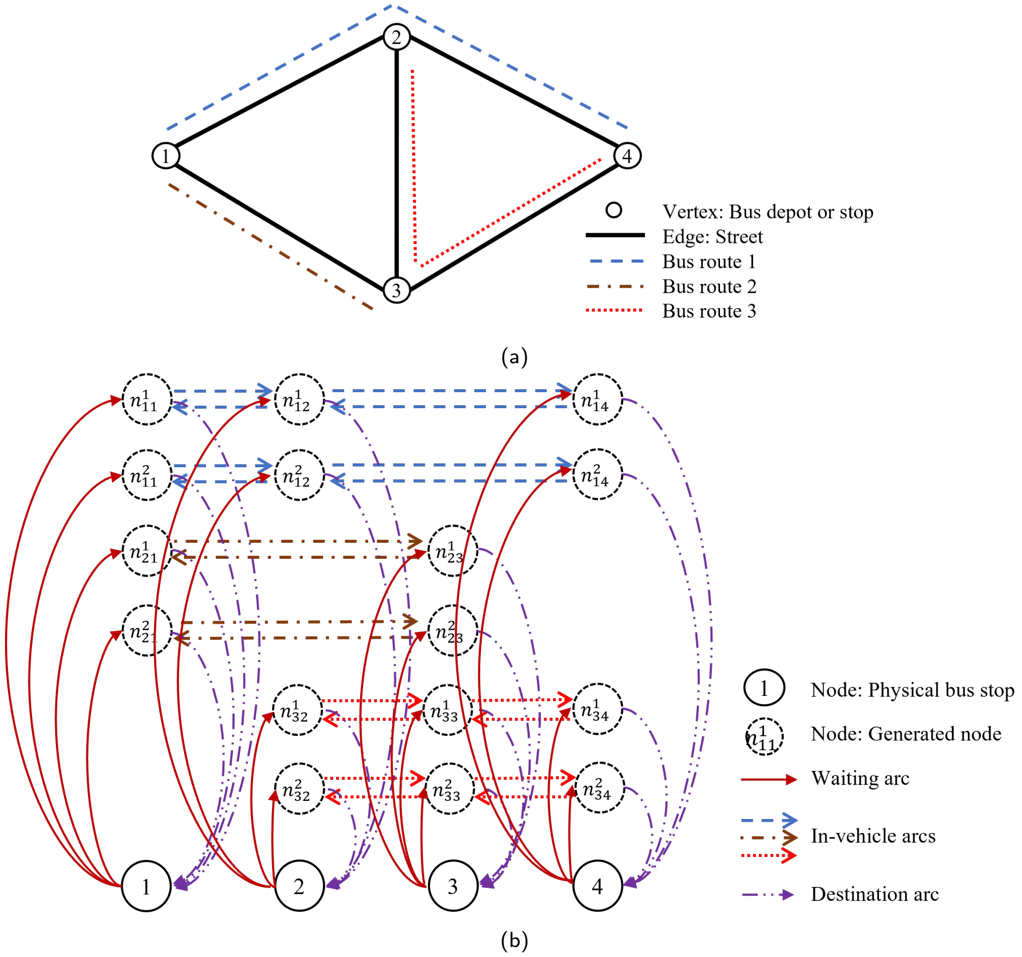


Figure 3.1: Example of the relation between (a) Physical road graph and (b) Extended trajectory graph

To facilitate the model formulation, the following assumptions are given.

- The bus route pool R is determined in advance. To satisfy all demand and reduce transfers, we ensure at least one direct route passing each OD pair when generating the possible bus routes. The passengers may transfer between different routes when the direct route takes much more time.

- For any route, buses depart in both directions at the same time and only one type of bus is selected.

- Buses travel as follows: starting from the depot to the route's first stop, going through multiple round trips, and finally returning to the depot from the first stop where service begins. The number of round trips depends on the total service time of each bus and the round-trip travel time.

- Different types of buses have diverse speeds on the same arc; that is, when passing through the same arc, their travel costs are distinct.
- The vehicle range is ignored in our model, regardless of the energy source used by the bus. Thus, our model may be more applicable to small and medium-sized cities without long-distance bus routes.
- The demand in the model is fixed.
- For simplicity, the bus capacity is not considered. The optimal strategies assignment model is not suitable for the congested transit system. Such an assignment provides the expected waiting time for the first bus arriving at the bus stop. If the capacity is considered, the passengers may fail to board the first full bus, which increases the waiting time. For capacity-constrained transit assignment, readers can refer to [22,27,57].

3.2.2 Notation

Sets and indices

- $h \in H$: bus types, $H = \{0 : \text{human-driven}, 1 : \text{autonomous}\}$
- $r \in R$: potential routes
- $i \in I$: OD pairs
- $j \in J \subseteq V$: potential bus depot location sites
- $e \in E$: edges of physical network
- $e \in E_r$: edges of route r
- $n \in N$: nodes of trajectory network
- $a \in A = A^W \cup A^I \cup A^D$: arcs of trajectory network including waiting arcs, in-vehicle travel arcs, and destination arcs; each arc a corresponds to a specific route and bus type, denoted by $r(a)$ and $h(a)$.

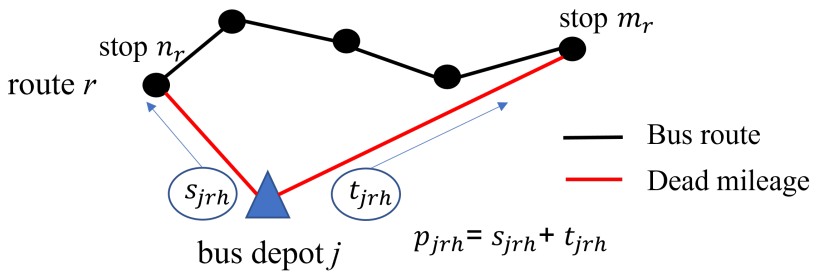
- $a \in A_n^+$ (A_n^-): outgoing (incoming) arcs from (to) node n
- $a \in A_n^{W+}$: outgoing waiting arcs from node n

Parameters

- FC_j: cost of a bus depot at site j
- SC: cost of each bus stop
- VC_h: cost of each h-type bus
- UC_h: unit operating cost of each h-type bus
- OC_{jrh}: travel time spent on dead mileage between depot j and one endpoint of route r for each h-type bus
- DC_{jrh}: travel time spent on dead mileage between depot j and the other endpoint of route r for each h-type bus
- RC_{rh}: travel time spent on route r for each h-type bus, equal to the product of single round trip travel time and the number of round trips
- c_a: travel cost on arc a
- κ_{jh}: depot capacity for h-type buses at site j
- D: number of drivers available
- M: a large number
- λ: weight
- δ_i: demand of OD pair i
- b_{ni}: net flow of OD pair i at node n; for OD pair i, if $n = O_i$, $b_{ni} = \delta_i$, if $n = D_i$, $b_{ni} = -\delta_i$, otherwise $b_{ni} = 0$.
- $F_{\max}, F_{\min}$: upper and lower bounds of the frequency

Variables

- $x_j \in \{0, 1\}$: binary variable, if the bus depot is located at site j, the value is 1; otherwise 0
- $y_r \in \{0, 1\}$: binary variable, if the bus route r is selected, the value is 1; otherwise 0
- $z_{rh} \in \{0, 1\}$: binary variable, if h-type buses are used on route r, the value is 1; otherwise 0



RC_{rh} = number of round trips $*$ round-trip time between $n_r$and m_r for h-type buses
OC_{jrh} = 2 $*$ shortest travel time between j and n_r for h-type buses
DC_{jrh} = 2 $*$ shortest travel time between j and m_r for h-type buses

Figure 3.2: Illustration of each part of operating time

- $f_r \in \mathbb{R}_+$: frequency of route r
- $s_{jrh} \in \mathbb{R}_+$: number of h-type buses allocated to one endpoint of route r from depot j
- $t_{jrh} \in \mathbb{R}_+$: number of h-type buses allocated to the other endpoint of route r from depot j
- $p_{jrh} \in \mathbb{R}_+$: number of h-type buses allocated to route r from depot j, and $p_{jrh} = s_{jrh} + t_{jrh}$
- $v_{ai} \in \mathbb{R}_+$: passenger volume on arc a for OD pair i
- $w_{ni} \in \mathbb{R}_+$: expected waiting time multiplied by the demand at node n for OD pair i

3.2.3 Model

As explained in Subsection 3.2.1, we aim to minimize the costs of operators and users. The operators' costs have two components: the capital cost C_1 and the operating cost C_2. In Equation (3.1), the three terms are respectively the costs of bus depots, bus stops, and all buses.

$$C_1 = \sum_{j\in J} FC_j x_j + SC \sum_{r\in R} NS_r y_r + \sum_{h\in H} VC_h \sum_{r\in R} \sum_{j\in J} p_{jrh} \tag{3.1}$$

The operating cost is the product of unit operating cost and operating time, see Equation (3.2). The operating time for each bus includes the travel time spent on route to provide the service and the travel time wasted on dead mileage. Figure 3.2 illustrates the parameters related to the operating time.

$$C_2 = \sum_{h \in H} UC_h \sum_{r \in R} \sum_{j \in J} (RC_{rh} p_{jrh} + OC_{jrh} s_{jrh} + DC_{jrh} t_{jrh}) \tag{3.2}$$

The users' costs C_3 is the total travel time of passengers, containing in-vehicle travel time and waiting time.

$$C_3 = \sum_{i \in I} \sum_{a \in A} c_a v_{ai} + \sum_{i \in I} \sum_{n \in N} w_{ni} \tag{3.3}$$

The detailed formulation of Model I is as below.

$$\min \lambda(C_1 + C_2) + (1 - \lambda) C_3 \tag{3.4a}$$

subject to:

$$\sum_{j \in J} s_{jrh} \geq f_r z_{rh} \sum_{e \in E_r} c_{eh}, \qquad \forall r \in R, h \in H \tag{3.4b}$$

$$\sum_{j \in J} t_{jrh} \geq f_r z_{rh} \sum_{e \in E_r} c_{eh}, \qquad \forall r \in R, h \in H \tag{3.4c}$$

$$p_{jrh} = s_{jrh} + t_{jrh}, \qquad \forall j \in J, r \in R, h \in H \tag{3.4d}$$

$$\sum_{r \in R} p_{jrh} \leq \kappa_{jh} x_j, \qquad \forall j \in J, h \in H \tag{3.4e}$$

$$\sum_{j \in J} p_{jrh} \leq M z_{rh}, \qquad \forall r \in R, h \in H \tag{3.4f}$$

$$\sum_{r \in R} \sum_{j \in J} p_{jr0} \leq D \tag{3.4g}$$

$$\sum_{h \in H} z_{rh} = y_r, \qquad \forall r \in R \tag{3.4h}$$

$$\sum_{a \in A_n^+} v_{ai} - \sum_{a \in A_n^-} v_{ai} = b_{ni}, \qquad \forall n \in N, i \in I \tag{3.4i}$$

$$v_{ai} \leq f_{r(a)} w_{ni}, \qquad \forall a \in A_n^{W+}, n \in N, i \in I \tag{3.4j}$$

$$v_{ai} \leq \delta_i z_{r(a)h(a)}, \qquad \forall a \in A^W, i \in I \tag{3.4k}$$

$$F_{\min} y_r \leq f_r \leq F_{\max} y_r, \qquad \forall r \in R \tag{3.4l}$$

$$p_{jrh} \in \mathbb{R}_+, \qquad \forall j \in J, r \in R, h \in H \tag{3.4m}$$

$$s_{jrh} \in \mathbb{R}_+, \qquad \forall j \in J, r \in R, h \in H \tag{3.4n}$$

$$t_{jrh} \in \mathbb{R}_+, \qquad \forall j \in J, r \in R, h \in H \tag{3.4o}$$

$$v_{ai} \in \mathbb{R}_+, \qquad \forall a \in A, i \in I \tag{3.4p}$$

$$w_{ni} \in \mathbb{R}_+, \qquad \forall n \in N, i \in I \tag{3.4q}$$

$$f_r \in \mathbb{R}_+, \qquad \forall r \in R \tag{3.4r}$$

$$x_j \in \{0,1\}, \qquad \forall j \in J \tag{3.4s}$$

$$y_r \in \{0,1\}, \qquad \forall r \in R \tag{3.4t}$$

$$z_{rh} \in \{0,1\}, \qquad \forall r \in R, h \in H \tag{3.4u}$$

The objective (3.4a) is to minimize the weighted sum of operators' costs and users' costs. Constraints (3.4b) and (3.4c) ensure that there are enough buses in both directions of each route. The term on the right-hand side is the number of h-type buses required in one direction of route r. Equations (3.4d) mean h-type buses allocated to route r from depot j are used in two directions of route r, as shown in Figure 3.2. In constraints (3.4e), the number of h-type buses allocated to all routes from each depot is restricted by the capacity of that depot. Constraints (3.4f) state that h-type buses for route r are unnecessary unless the route is chosen and determined to deploy h-type buses. Constraint (3.4g) represents that the fleet size of human-driven buses cannot exceed the number of drivers available. Constraints (3.4h) guarantee that each route used corresponds to only one type of vehicle. Constraints (3.4i) and (3.4j) assign the passenger flows. Appendix A provides a more detailed explanation. Constraints (3.4k) state users can only use the routes and bus types that are part of the solution; $r(a)$ and $h(a)$ respectively refer to the route and bus type corresponding to arc a. Constraints (3.4l) limit the value range of the frequencies. The remaining constraints (3.4m)–(3.4u) are the variable restrictions.

3.2.4 Linear approximation with the method by Cancela et al. [16]

Due to the constraints (3.4b), (3.4c), and (3.4j), the proposed model (3.4) is obviously nonconvex and nonlinear, making the problem difficult to solve. Therefore, we take advantage of the approach developed by Cancela et al. [16] to linearize and approximate

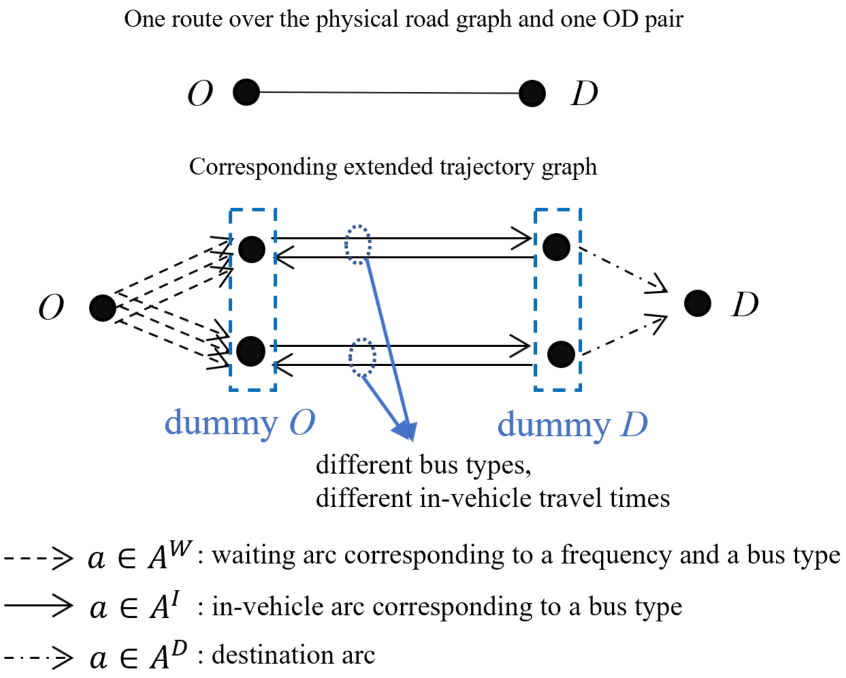

One route over the physical road graph and one OD pair
O
D
Corresponding extended trajectory graph
O
D
dummy O
dummy D
different bus types,
different in-vehicle travel times
$a \in A^W$: waiting arc corresponding to a frequency and a bus type
$a \in A^I$: in-vehicle arc corresponding to a bus type
$a \in A^D$: destination arc

Figure 3.3: Enumerate all possible frequencies and bus types in the trajectory graph

our model. As with [16], a new discrete set Θ is utilized to represent possible values of frequencies and $\Theta = \{\theta_f, f \in F\}$ where f is the index. Then, variable z_{rh} is substituted by z_{rfh} that takes value 1 if route r has frequency θ_f and uses bus type h, 0 otherwise. With parameter θ_f and variable z_{rfh}, the linearized model is written by problem (3.5). Note that $\theta_{f(a)}$ is constant, where $f(a)$ means the frequency index in set Θ corresponding to the wait arc $a \in A^W$. To be more specific, we need to modify the extended trajectory graph by creating wait arcs for all possible frequencies in Θ. An instance for $|F| = 3$ and $|H| = 2$ is depicted in Figure 3.3. In fact, only one waiting arc in this graph will be enabled due to constraints (3.5e) and (3.5g).

$$\min \lambda(C_1 + C_2) + (1 - \lambda)C_3 \tag{3.5a}$$

subject to:

$$\sum_{j \in J} s_{jrh} \geq \sum_{f \in F} \theta_f z_{rfh} \sum_{e \in E_r} c_{eh}, \qquad \forall r \in R, h \in H \tag{3.5b}$$

$$\sum_{j\in J} t_{jrh} \geq \sum_{f\in F} \theta_f z_{rfh} \sum_{e\in E_r} c_{eh}, \qquad \forall r \in R, h \in H \tag{3.5c}$$

$$\sum_{j\in J} p_{jrh} \leq M \sum_{f\in F} z_{rfh}, \qquad \forall r \in R, h \in H \tag{3.5d}$$

$$\sum_{h\in H}\sum_{f\in F} z_{rfh} = y_r, \qquad \forall r \in R \tag{3.5e}$$

$$v_{ai} \leq \theta_{f(a)} w_{ni}, \qquad \forall a \in A_n^{W+}, n \in N, i \in I \tag{3.5f}$$

$$v_{ai} \leq \delta_i z_{r(a)f(a)h(a)}, \qquad \forall a \in A^W, i \in I \tag{3.5g}$$

$$z_{rfh} \in \{0,1\}, \qquad \forall r \in R, f \in F, h \in H \tag{3.5h}$$

(3.4d), (3.4e), (3.4g), (3.4i), (3.4m)–(3.4q), (3.4s), and (3.4t)

3.3 Path-based model (Model II)

Although the BDL&TNDP becomes linear and can provide the optimal solution, it is still burdensome to compute the global optimum when the size of the road graph is large [16]. A slight increase in the road network size leads to a tremendous expansion of the pool of candidate routes, thus enlarging the size of the trajectory network significantly. Therefore, we propose a new simplified path-based model (Model II) that assigns passengers heuristically without using the trajectory network. We use the model to obtain strategic decisions and assess the their performance in Model I.

3.3.1 Problem setting and assumptions

The main complexity of BDL&TNDP lies in the TNDP, which includes a node-based passenger assignment to model the passengers' behaviors. Although the transfers are allowed in Model I, the penalty cost of transfer tends to prevent transferring between different routes. Therefore, most users are direct travelers who take one bus route to reach the destination. Unless the direct distance is too long, some may prefer transfers. Motivated by this situation, we only consider the costs of direct travelers (in-vehicle time and waiting time) in the part of users' costs of Model II. In such a model, we predefine each OD pair's candidate direct travel paths for passenger assignment and assume an equivalent expected waiting time of users who take the same route. Instead of exactly assigning the passenger flows according to route frequencies and computing the total waiting time at each node in Model I, the passenger assignment in the simplified model is based on the default paths and independent of the route frequencies, which is a heuristic approximation. However, there is no need to construct the trajectory graph in such a model, so the resulting number of variables and constraints can become much fewer.

Besides the assumptions in Subsection 3.2.1, two new assumptions are introduced here to formulate the model.

- The candidate direct travel paths of passengers are generated by the k-shortest path algorithm [97].
- The expected wait time for a line is the same for all passengers on that line and is proportional to the headway of that line (the inverse of the frequency).

3.3.2 Additional notation

Sets and indices

- $q \in Q_i$: travel paths of OD pair i, including k-shortest paths

Parameters

- l_{qh}: travel time on path q using h-type bus
- Δ_{qr}: constant value, if path q is a part of route r, the value is 1; 0 otherwise
- α: coefficient for estimating waiting time

Variables

- $u_{qrhi} \in \mathbb{R}_+$: number of direct travelers for OD pair i who choose path q along route r using h-type buses
- $\phi_r \in \mathbb{R}_+$: total number of passengers who board route r

3.3.3 Formulation

Due to the change in assigning passengers, a new function expressing users' costs is displayed in Equation (3.6). The first term represents the total travel time of direct travelers; the second term is the total waiting time. In this formula, set Q_i and coefficient α can be adjusted to obtain different solutions. Set Q_i is given by k-shortest path method, so we can choose different k to change it. More intuitively, C_3' is an approximation of C_3, and we use different k and different α to investigate approximation performance.

$$C_3' = \sum_{i \in I} \sum_{q \in Q_i} \sum_{r \in R} \sum_{h \in H} l_{qh} u_{qrhi} + \alpha \sum_{r \in R} \frac{\phi_r}{f_r} \tag{3.6}$$

With the simplified assignment method, we can rewrite Model I into the following problem.

$$\min \lambda(C_1 + C_2) + (1 - \lambda) C_3' \tag{3.7a}$$

subject to:

$$u_{qrhi} \leq \Delta_{qr} \delta_i z_{rh}, \qquad \forall i \in I, q \in Q_i, r \in R, h \in H \tag{3.7b}$$

$$\phi_r = \sum_{i \in I} \sum_{q \in Q_i} \sum_{h \in H} u_{qrhi}, \qquad \forall r \in R \tag{3.7c}$$

$$\sum_{q \in Q_i} \sum_{r \in R} \sum_{h \in H} u_{qrhi} = \delta_i, \qquad \forall i \in I \tag{3.7d}$$

$$u_{qrhi} \in \mathbb{R}_+, \qquad \forall i \in I, q \in Q_i, r \in R, h \in H \tag{3.7e}$$

$$\phi_r \in \mathbb{R}_+, \qquad \forall r \in R \tag{3.7f}$$

(3.4b)–(3.4h), (3.4l)–(3.4o), and (3.4r)–(3.4u)

The objective function (3.7a) is to minimize the weighted sum of operators' costs and approximate users' costs. Constraints (3.7b) restrict the number of passengers of OD pair i who select direct path q by using route r and h-type bus. Constraints (3.7c) compute the total passenger flow volume boarding each route. Constraints (3.7d) ensure the number of assigned passengers for each OD pair is always equal to that OD demand.

3.3.4 Linear approximation

Similar to Model I, Model II has a nonlinear term ϕ_r/f_r. To handle it, we introduce an additional variable $\pi_r \in \mathbb{R}_+$ such that $\pi_r f_r = \phi_r$, meanwhile, approximate the continuous variable f_r with the discrete set $\Theta = \{\theta_f, f \in F\}$ where f is the index of set and the binary variable z_{rfh} like Subsection 3.2.4. Finally, the following model (3.8) can be obtained.

$$\min \lambda(C_1 + C_2) + (1-\lambda)C_3'' \tag{3.8a}$$

where

$$C_3'' = \sum_{i \in I} \sum_{q \in Q_i} \sum_{r \in R} \sum_{h \in H} l_{qh} u_{qrhi} + \alpha \sum_{r \in R} \pi_r$$

subject to:

$$\pi_r \sum_{f \in F} \sum_{h \in H} \theta_f z_{rfh} = \phi_r \qquad \forall r \in R \tag{3.8b}$$

$$u_{qrhi} \le \Delta_{qr} \delta_i \sum_{f \in F} z_{rfh}, \qquad \forall i \in I, q \in Q_i, r \in R, h \in H \tag{3.8c}$$

(3.4d), (3.4e), (3.4g), (3.4m)–(3.4o), (3.4s), (3.4t),

(3.5b)–(3.5e), (3.5h), and (3.7c)–(3.7f)

As shown in constraints (3.8b), the real variable f_r is replaced by $\sum_{f \in F} \theta_f z_{rfh}$ so that the left term includes the products of continuous and binary variables. This unique structure can be linearized easily. The resulting equivalent linear constraints of (3.8b) are given as follows.

$$\sum_{f \in F} \sum_{h \in H} \theta_f \Pi_{rfh} = \phi_r, \qquad \forall r \in R$$

$$\Pi_{rfh} \le M z_{rfh}, \qquad \forall r \in R, f \in F, h \in H$$

$$\Pi_{rfh} \le \pi_r, \qquad \forall r \in R, f \in F, h \in H$$

$$\Pi_{rfh} \ge \pi_r - M(1 - z_{rfh}), \qquad \forall r \in R, f \in F, h \in H$$

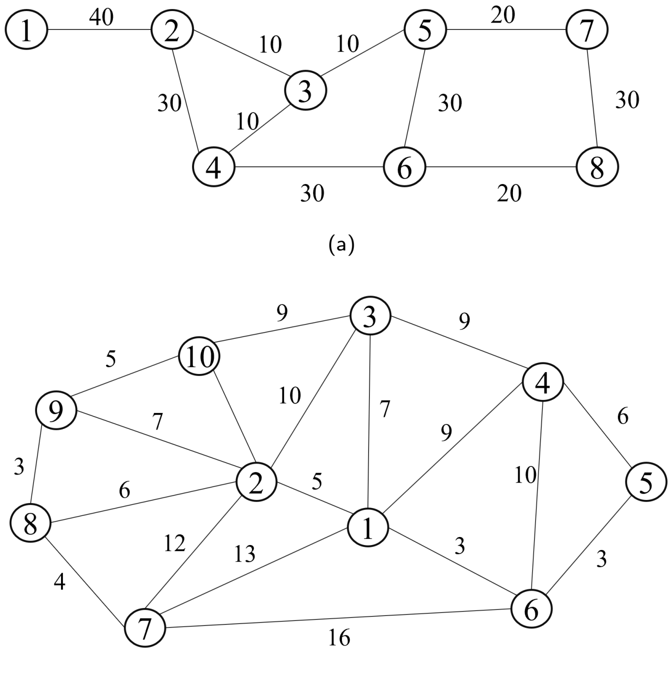
$$\Pi_{rfh} \ge 0, \qquad \forall r \in R, f \in F, h \in H$$

where $\Pi_{rfh} \in \mathbb{R}_+$ is the auxiliary variable.

3.4 Experiments and results

3.4.1 Parameter settings

In order to evaluate the performance of the two models (linearized Models I and II), we implemented two networks from [16] and [91] to conduct the experiments, as depicted in Figure 3.4. These two networks were also applied in [16,31]. The numbers next to the edges are the in-vehicle travel times of human-driven buses, expressed in minutes. We assumed that the in-vehicle travel time on an edge for autonomous buses is a multiple of that for human-driven buses, denoted by m, such that $c_{e1} = mc_{e0}$. Detailed OD demand data can be found in the papers listed above. Different unit of demand quantity (e.g., trips per hour [91]) was converted to trips per minute. Some basic information about the networks is provided in Table 3.1.



(a)

(b)

Figure 3.4: Networks: (a) Network 1 [16]; (b) Network 2 [91]

Table 3.1: Basic information of the networks used

Network	# of vertices	# of edges	# of OD pairs	total demand quantity (trips/min)
1	8	10	4	4.00
2	10	19	9	61.67

Because the sizes of the networks and their demands are different, we set parameters for them separately. However, some parameters used the same value or were set in the same way. The parameters with the same value are presented in Table 3.2. The set of depot location candidates (J) for each network includes all nodes in it. As for the operating cost, the total service time (RC_{rh}) and the dead mileage (O_{jrh} and D_{jrh}) were determined by using the equations in Figure 3.2. RC_{rh} can be expressed by

$$RC_{rh} = \frac{W_r}{T_{rh} + B_h} \times T_{rh}, \forall h \in H$$

where W_r is the operating time of route r; T_{rh} is the round-trip time of route r with h-type buses, $T_{r1} = mT_{r0}$; B_h is the drivers' break time, we set $B_0 = T_{r0}/6$ and $B_1 = 0$.

Table 3.2: Parameters with the same value in the networks

Parameters (Unit)	Value
Bus type set H	{0: Human-driven 1: Autonomous}
Cost of bus stop SC (¥/min)	0.03
Cost of bus VC_h (¥/min)	$VC_0 = VC_1 = 6.08$
Unit operating cost UC_h (¥/min)	$UC_0 = 20$; $UC_1 = 10$
Multiple m	1.5

Next, we explain the parameters depending on the network. One important part is the candidate route pool R. Similar to [16], the following ways were applied to generate the route pools. For network 1, we generated all possible routes by a modified depth-first search in [80] and removed the routes whose at least one endpoint that is neither origin nor destination of any OD pair. The resulting R contains 79 routes. For Network 2, it is difficult to list all possible routes. We applied the k-shortest path method to compute k-shortest paths for each OD pair. Then, we set $k = 5$ and obtained 45 bus routes. In Model

II, the possible paths of passengers Q_i for OD pair i were also determined by k-shortest path algorithm. In contrast to the one-time determination of the candidate route pool, we considered different k to determine multiple path sets Q_i in Model II and evaluated the corresponding performance. For both networks, $k \in \{1, 2, 3, 4, 5\}$. Besides, other constant parameters are summarized in Table 3.3. It is assumed that the capacity and cost of the bus depot are the same for each location $j \in J$. Note that the demand in the data is expressed in terms of trips per minute. Thus, we used yen per minute to expresse the users' costs and the operators' costs. The cost of a bus depot was assumed to be 50 million yen for Network 1 and 100 million yen for Network 2. If the service life of the bus depot is 15 years, the cost can be converted to a cost per minute as shown in the Table 3.3.

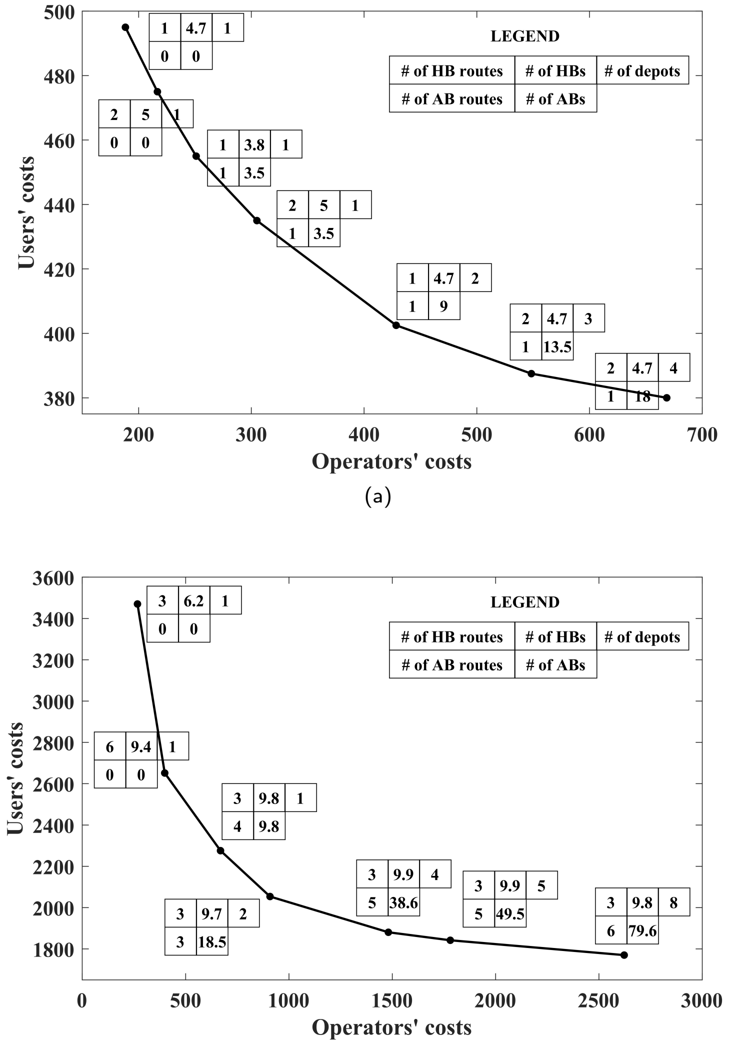
The models were implemented using Gurobi Optimizer 9.0.1 in the Python environment on an Intel i7-8700 CPU with 12 cores and 16GB of RAM. The modified depth-first search and k-shortest path methods were realized by using the NetworkX package [41] in Python.

Table 3.3: Parameters setting different values for the networks

Networks	Parameters	Value
1	Capacity of bus depot κ_{jh}	κ_{j0}=10, κ_{j1}=5
	Cost of bus depot FC_j (¥/min)	10.32
	Number of drivers D	5
	Frequency set Θ (trips/min)	$\{1/60, 1/30, 1/5\}$
2	Capacity of bus depot κ_{jh}	κ_{j0}=20, κ_{j1}=10
	Cost of bus depot FC_j (¥/min)	20.64
	Number of drivers D	10
	Frequency set Θ (trips/min)	$\{1/20, 3/20, 1/4, 1/3\}$

3.4.2 Optimization results of Model I

We solved the approximately linear Model I for both networks with the given parameters. By using different weights λ, we obtained the efficient frontiers of operators' and users' costs in Figure 3.5. Next to each node in the figures, we also presented the detailed results of the numbers of human-driven bus (HB) routes and autonomous bus (AB) routes, the numbers of human-driven buses (HBs) and autonomous buses (ABs), and the number of bus depots. A noticeable trend for both frontiers is that the users' costs will decrease with the increment of operators' costs. With the limitation on the number of drivers, users will be inconvenient to travel without autonomous buses, even though operators can reduce large expenditures (see first and second nodes). As operators provide enough funding to introduce more autonomous buses, users' travel costs will be greatly reduced. Therefore, autonomous buses would be a good option if operators want to build a user-centric bus transit system with limited drivers.



(a)

(b)

Figure 3.5: Efficient frontiers of operators' and users' costs: (a) Network 1; (b) Network 2

Then, as shown in Tables 3.4 and 3.5, we considered the different numbers of drivers available (i.e., D) and analyzed their effects on two networks. There are some similar observations for these networks. Without available drivers ($D = 0$), both operators' and users' costs are at a higher level. Increasing the number of drivers available reduces the operators' and users' costs in general. When the weight λ is 0.5, the operators' costs will sometimes rise with the increase of D, but the users' costs can keep reducing. Moreover, looking at the results when $\lambda = 0.9$, we found 5 and 10 bus drivers can satisfy the demand of Network 1 and Network 2, respectively, and that more available drivers do not affect the results.

Table 3.4: Effect of the number of drivers available on Network 1

λ	D	Operators' costs	Users' costs
0.1	0	633.8	421.5
	5	548.4	387.5
	10	463.6	340.0
0.5	0	243.6	532.5
	5	188.2	495.0
	10	258.3	415.0
0.9	0	194.1	622.5
	5	188.2	495.0
	10	188.2	495.0

Table 3.5: Effect of the number of drivers available on Network 2

λ	D	Operators' costs	Users' costs
0.1	0	2458.8	1931.9
	10	1780.8	1841.9
	20	1557.8	1557.8
0.5	0	823.5	2494.7
	10	597.8	2313.8
	20	819.9	1914.3
0.9	0	276.1	3950.0
	10	267.1	3470.0
	20	267.1	3470.0

3.4.3 Evaluating Model II

In this part, we first compared two models regarding the model size and the computational efficiency. Then, we evaluated the solutions of Model II with different Q_i (i.e., different k) and α by substituting them into Model I to obtain the objective values. The results were compared with the optimal values of the original Model I.

Table 3.6 exhibits the numbers of variables and constraints for the two models with different networks. Obviously, the size of Model II is smaller than that of Model I no matter the number of variables or the number of constraints. As the number of travel candidate paths (k) of passengers rises, the numbers of variables and constraints of Model II increase linearly, but they are still much fewer than those of Model I for both networks.

Table 3.6: Comparison of the size of the two models

	Networks	Model I	Model II				
			$k=1$	2	3	4	5
# of Variables	1	32641	5625	6257	6889	7521	7995
	2	49249	4385	5195	6005	6815	7625
# of Constraints	1	23818	4208	4840	5472	6104	6578
	2	39756	3360	4170	4980	5790	6600

After analyzing the scales of Models I and II, we compared the optimization results of the two models in terms of the gap between the upper and lower bounds and the computing time, as presented in Table 3.7. We conducted the comparative experiments of both networks and set the weights $\lambda \in \{0.1, 0.5, 0.9\}$ for the two models. Moreover, different path numbers k for generating Q_i and waiting time coefficients α in Model II were used to obtain the solution for evaluation, where $k \in \{1, 2, 3, 4, 5\}$ and $\alpha \in \{0, 0.3, 0.6, 0.9, 1.2, 1.5\}$. Concerning each k in Model II, the average gap and the average computing time for different α are shown in this table. For all experiments, the time limit is 3600s.

From Table 3.7, we see that the BDL&TNDP on Network 1 can be solved in a short time by these two models. In contrast, Model II converges faster than Model I due to fewer variables and constraints. On Network 2, the superiority of Model II is more obvious. Model I runs out the prespecified time with the gaps more than 6% when $\lambda = 0.1$ and 0.5. However, Model II can get the optimal solutions within a few seconds. As the weights increase (i.e., less attention to the user's cost), it takes less time to solve the problem on both networks, which indirectly suggests that TNDP is the main cause of the difficulty of BDL&TNDP.

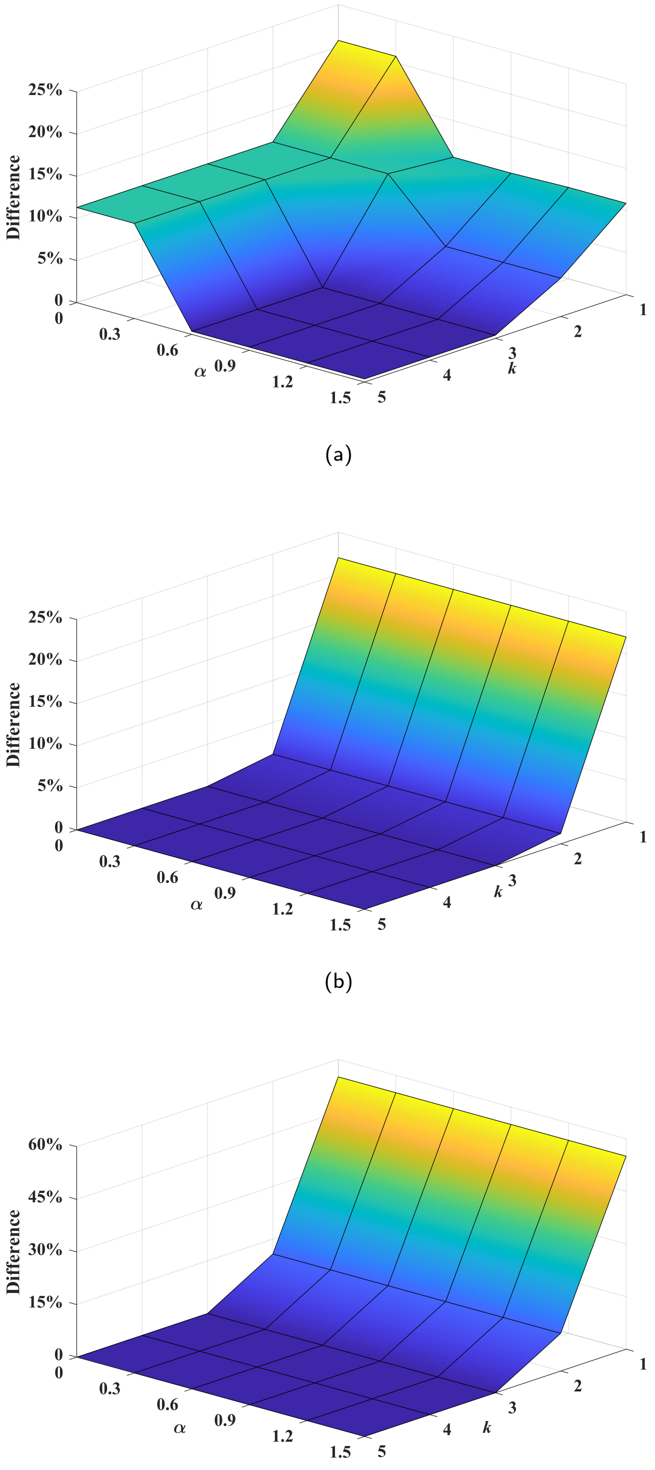
Table 3.7: Comparison of computational efficiency

		Gap(%)						Time(s)					
λ	Networks	Model I	Model II					Model I	Model II				
			$k = 1$	2	3	4	5		$k = 1$	2	3	4	5
0.1	1	0	0					9.73	0.49	0.77	1.23	1.35	1.56
	2	6.2	0					3600	0.81	2.60	6.09	7.17	11.82
0.5	1	0	0					0.94	0.24	0.27	0.50	0.48	0.57
	2	12	0					3600	0.42	0.97	1.87	1.87	6.38
0.9	1	0	0					0.31	0.20	0.20	0.22	0.23	0.24
	2	0	0					128	0.20	0.61	0.57	0.56	0.77

So far, we have only explicated the scale of Model II and the speed at which it is used to solve BDL&TNDP. In order to evaluate the quality of solutions (binary variables: x_j, y_r, and z_{rfh}) of Model II, we substituted them into Model I and optimized new Model I to achieve the optimal objective values. The obtained objective values were compared with those from the original Model I by direct optimization, and the differences between them were counted as percentages. The smaller the difference, the better the solution. As mentioned above, various combinations of k and α were deployed in Model II to get the solutions for evaluation. Figures 3.6 and 3.7 illustrate the differences between the objective values of new Model I and original one for the selected weights λ on Networks 1 and 2, respectively.

As seen in Figure 3.6, the differences are sensitive to both k and α when λ is smaller. Increasing the number of candidate paths of passengers and the estimated waiting time can efficiently reduce the differences. Nevertheless, the differences barely changes and stay at a low level ($\leq 0.4\%$) when k is greater than 3, and α is greater than 0.6. As λ grows, the adjustments of α have nothing to do with the differences, but more candidate paths can help receive the solutions with zero differences. Three candidate paths seem to be enough for Model II to obtain excellent solutions.

From the computing time in Table 3.7, it is suggested that the problem for Network 2 is more complex than that for Network 1. However, Model II still performs well and provides satisfactory solutions for Network 2, as shown in Figure 3.7. Turning now to Figure 3.7, we can make some similar observations. When $\lambda = 0.1$, the best difference around 2% can be found at the points where $k \geq 3$ and $\alpha = 0.9$. In Figure 3.7b with $\lambda = 0.5$, changing the number of candidate paths k has no impact on the differences. For each α, the differences remain the same value even though k is increased. Most parameter combinations can provide a good solution whose differences are within 4%. By contrast, the difference can be decreased if more k is considered when $\lambda = 0.9$. In addition, we note that the differences become worse as λ rises on Network 2. For $\lambda = 0.9$, the differences are between 20% and 25%. However, it only takes less than 1s for Model II to provide the feasible solutions, while Model I cannot obtain any solution in such a short time. Further, it is satisfactory that the performance of the solution is better if more attention is paid to the users' costs in the objective function since the path-based model is used to simplify the passenger assignment corresponding to the users' costs in BDL&TNDP.

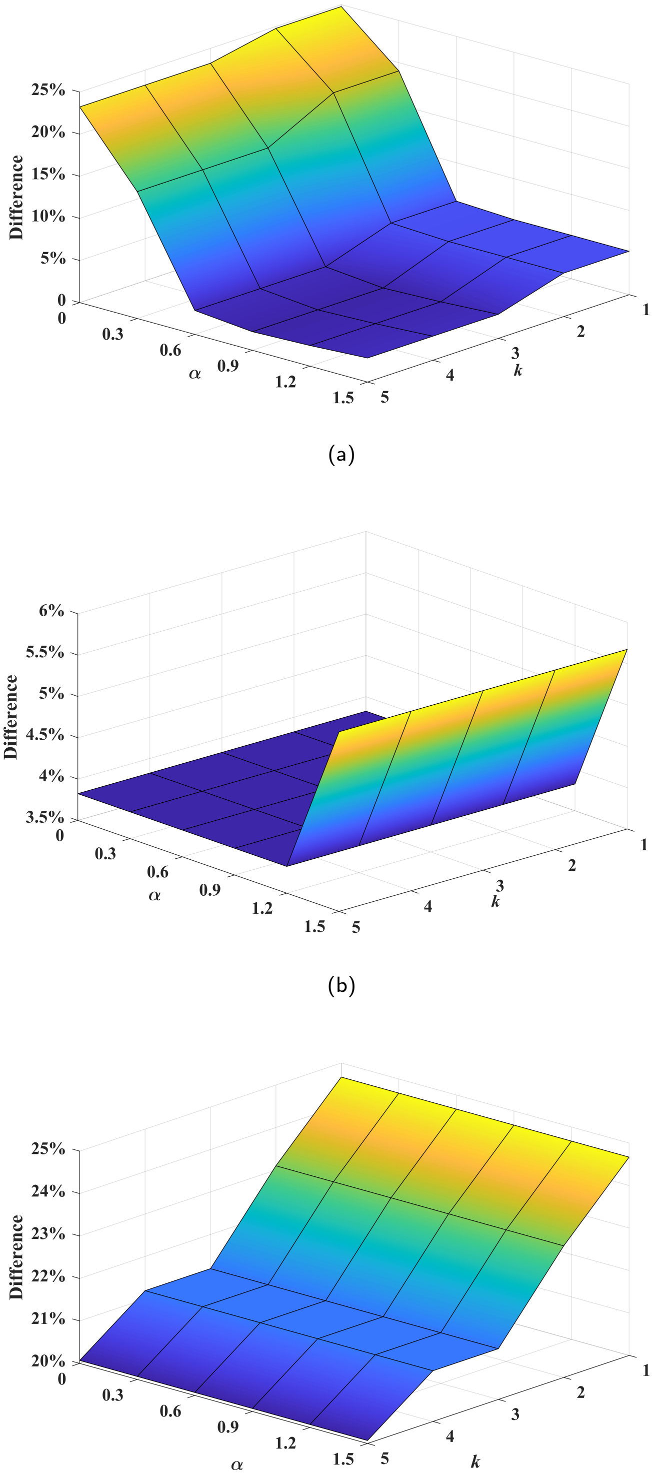


(a)

(b)

(c)

Figure 3.6: Resulting differences on Network 1: (a) $\lambda = 0.1$; (b) $\lambda = 0.5$; (c) $\lambda = 0.9$



(a)

(b)

(c)

Figure 3.7: Resulting differences on Network 2: (a) $\lambda = 0.1$; (b) $\lambda = 0.5$; (c) $\lambda = 0.9$

3.5 Summary

This chapter proposes an MINLP model (Model I) for the BDL&TNDP in the emerging bus transit systems operating with human-driven and autonomous buses considering the limited drivers. To solve the problem, we linearized the model approximately by using the method in [16]. Even so, handling the linearized problem is still time-consuming because of the complexity of passenger assignments. Therefore, we developed a path-based model (Model II) without the consideration of exactly assigning the passenger and computing the waiting time at each node. By contrast, the passengers are assigned to a set of predefined paths, and the waiting time of each route is estimated. In this way, we can reduce many variables and constraints and save computing time. For evaluating the proposed path-based Model II, we substituted the solutions from linearized Model II with different parameters into linearized Model I to obtain the objective values. The resulting values were compared with those of the direct optimization of Model I and the differences between them were calculated.

The experiments were conducted with two networks from previous studies. We obtained the efficient frontiers of operators' and users' costs for the networks with the approximately linear Model I and the given parameters. The results show that the users' costs are high without autonomous buses and under the restriction of drivers. However, if more autonomous buses were introduced, the users could save considerable travel times, revealing the importance of autonomous buses in the emerging bus transit systems. We also compared the linearized Models I and II with regard to the model size and computational efficiency. On both networks, we found that Model II can be solved optimally in a few seconds, which is much faster than Model I. Furthermore, the solutions from Model II have good performance in Model I. For Network 1, the solutions from Model II can offer zero-difference objective values. For Network 2, although the zero-difference objective value is difficult to receive, Model II presents reasonable solutions with more minor differences when more attention is paid to users' costs.

Chapter 4

Optimizing the strategic decisions for one-way station-based carsharing systems

4.1 Introductory remarks

In this chapter, we focus on the strategic design of one-way station-based carsharing systems from the vantage point of the system operator. The most related research (e.g., [9, 12, 15]) copes with strategic problems based on the traditional risk-neutral two-stage stochastic programming by considering the expectation value as the preference criterion. However, the resulting decisions may be poor under certain realizations of random data. For non-repetitive decision-making problems, such as location planning and network design, a risk-averse approach would provide more robust solutions [72]. Therefore, we analyze the downside risk, which refers to the financial risk of the actual return being below the expected return, in this study. With the objective of maximizing the return and minimizing the risk, we propose a two-stage risk-averse stochastic MINLP model, where the conditional value-at-risk (CVaR) is specified as the risk measure, to optimize strategic decisions involving station locations, station capacities, and fleet sizes for one-way station-based carsharing systems. With a training and testing method, we also show the advantage of the risk-averse model under stochastic demand. In order to solve the two-stage stochastic MINLP model, two customized methods are developed: a branch-and-cut algorithm and a scenario decomposition algorithm. The model and algorithms are verified on scenario demand data generated from historical data of Ha:mo RIDE Toyota in Japan.

The remainder of the chapter is as follows. Section 4.2 provides an elaborate model formulation for making strategic decisions. Section 4.3 is devoted to the two algorithms for solving the problem. Subsequently, we report computational results in Section 4.4 and conclude this study in Section 4.5. It is noted that this chapter has been published in Zhang et al. [101].

4.2 Mathematic formulation

4.2.1 Assumptions

In this study, we determine the station locations, station capacities, and fleet sizes to design one-way station-based carsharing system with a risk-averse model. To formulate the model, some underlying assumptions are as follows.

- The trip demand data are available or predictable in advance, including for different scenarios. Each scenario represents one possible day with a set of estimated trips. The probability is the same for each scenario. Each trip is a tuple of four elements: origin, destination, departure time, and arrival time. The price of each trip can be computed on the basis of the information above.

- For the trip demand to be satisfied, there must be vehicles available at the origin and parking spaces at the destination.

- The working time of the carsharing system is divided into equal time intervals (5 minutes), so we can gather together the trips with the same departure or arrival time interval. In this way, the problem size can be reduced.

- All stations have at least one parking space and have individual maximal capacities that depend on the local conditions. The station cost consists of land, construction, and charging-pile costs. The unit land costs vary depending on the location, but the unit construction and charging-pile costs are fixed.

- Considering the station and vehicle costs tend to be too high to get a profitable system, the profit return in our model only depends on the trip revenue and operating cost. Alternatively, the station and vehicle costs are restricted by a budget constraint. Çalık and Fortz [15] addressed a similar issue by introducing a cost factor to forcibly reduce the cost. This implies the operator must study the pricing policy and other income resources comprehensively, but this issue is out of the scope of this paper.

- The state of charge (SoC) of the EV's battery is ignored in this study. In the case of Ha:mo RIDE Toyota, the vehicles are mostly used for short trips, and a previous study [83] showed that no vehicle ever became unavailable in the system because of a low SoC. Therefore, the proposed model is more suitable for short-range carsharing systems to counter the last-mile problem.

- The system operator does not take into account operational activities (e.g., relocation, staff allocation, or staff scheduling) when making strategic decisions.

4.2.2 Notation

Before providing the detailed model formulation, we should explain the notation used in the model.

Sets and indices

- $s \in S$: scenarios
- $i \in I_s$: trips in scenario s
- $t \in T$: time intervals
- $j \in J$: potential location sites

Parameters

- $start_i$, $end_i \in T$: start- and end-intervals of trip i
- $origin_i$, $dest_i \in J$: origin and destination of trip i
- P_i^s: charging price for trip i in scenario s
- C_1: operating cost per parking space per scenario

- C_2: operating cost per vehicle per scenario
- H_j: cost for setting a parking space at site j, including land, construction, and charging-pile costs
- F: cost of purchasing a vehicle
- B: available budget
- M_j: maximum number of parking spaces that can be set at site j
- λ: weight value, ranging from 0 to 1
- β: confidence level

Variables

- $\alpha \in \mathbb{R}$: auxiliary variable for obtaining minimum β-CVaR
- $p_j \in \mathbb{Z}_+$: number of parking spaces at site j
- $v \in \mathbb{Z}_+$: number of vehicles in the system
- $n_{j,t}^s \in \mathbb{Z}_+$: number of vehicles at site j at the beginning of time interval t in scenario s
- $z_i^s \in \{0,1\}$: binary variable, if trip i in scenario s is served, the value is 1; otherwise 0

4.2.3 Model

In this part, we present the two-stage risk-averse stochastic MINLP model in the deterministic equivalent form. The first-stage constraints (4.1b) and (4.1c) are given by using variables α, v and p_j, that is, first-stage decision variables. The second-stage decision variables z_i^s and $n_{j,t}^s$, associated with scenario s, are restricted by the second-stage constraints (4.1d)–(4.1g).

$$\begin{aligned}\min \frac{\lambda}{|S|} \sum_{s\in S} & \left(-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v \right) \\ &+ (1-\lambda)\left(\alpha + \frac{1}{(1-\beta)|S|} \sum_{s\in S} \left[-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v - \alpha \right]_+ \right)\end{aligned} \tag{4.1a}$$

subject to:

$$\sum_{j \in J} H_j p_j + Fv \leq B \tag{4.1b}$$

$$p_j \leq M_j, \qquad \forall j \in J \tag{4.1c}$$

$$n^s_{j,t+1} = n^s_{j,t} - \sum_{\substack{i: origin_i = j \\ start_i = t}} z^s_i + \sum_{\substack{i: dest_i = j \\ end_i = t}} z^s_i, \qquad \forall s \in S, j \in J, t \in T \backslash \{t_{\text{last}}\} \tag{4.1d}$$

$$n^s_{j,t} \geq \sum_{\substack{i: orgin_i = j \\ start_i = t}} z^s_i, \qquad \forall s \in S, j \in J, t \in T \tag{4.1e}$$

$$p_j - n^s_{j,t} \geq \sum_{\substack{i: dest_i = j \\ end_i = t}} z^s_i, \qquad \forall s \in S, j \in J, t \in T \tag{4.1f}$$

$$\sum_{j \in J} n^s_{j,1} = v, \qquad \forall s \in S \tag{4.1g}$$

$$\alpha \in \mathbb{R} \tag{4.1h}$$

$$p_j \in \mathbb{Z}_+, \qquad \forall j \in J \tag{4.1i}$$

$$v \in \mathbb{Z}_+ \tag{4.1j}$$

$$z^s_i \in \{0, 1\}, \qquad \forall s \in S, i \in I_s \tag{4.1k}$$

$$n^s_{j,t} \in \mathbb{Z}_+, \qquad \forall s \in S, j \in J, t \in T \tag{4.1l}$$

The objective function (4.1a) minimizes the weighted sum of the measures of profitability (expected loss, i.e., the opposite of the expected return) and risk (CVaR). Note that $[x]_+ = \max\{x, 0\}$, for $x \in \mathbb{R}$. The loss function in each scenario is equal to the operating costs of the built parking spaces and purchased vehicles ($C_1 \sum_{j \in J} p_j + C_2 v$) reduced by the revenue of served trips ($\sum_{i \in I_s} P^s_i z^s_i$). In Appendix B, we present a detailed explanation of the objective function. Constraint (4.1b) states the costs for setting up parking spaces and purchasing cars are within the available budget. Constraints (4.1c) show the maximal capacity restriction at each potential location site. In addition, there are additional constraints depending on the scenario. Constraints (4.1d) are for vehicle flow conservation at each station for each time interval. The departure time and arrival time of each trip are known parameters in the model, so no additional constraints are required to represent the travel time. Constraints (4.1e) ensure there are enough vehicles for trips served at each station and each time interval. Constraints (4.1f) restrict the number of vehicles arriving at a station to the number of available parking spaces at that station during each time interval, and at the same time, ensure there are enough parking spaces for the vehicles at site j. Constraints (4.1g) indicate that the total number of vehicles in the system is always equal to the number of purchased vehicles. Constraints (4.1h)–(4.1l) are variable restrictions.

4.3 Solution methods

Due to plenty of variables and constraints, the strategic decision problem becomes more difficult when the number of scenarios increases. To deal with the problem efficiently, we present two solution methods including the branch-and-cut algorithm and scenario decomposition algorithm in this section. The former algorithm mainly focuses on the nonlinear CVaR function, while the latter one pays attention to the special block-angular structure of the stochastic problem.

4.3.1 Branch-and-cut algorithm

Takano et al. [87] proposed two cutting-plane algorithms to handle the CVaR function in a portfolio optimization problem with a nonconvex transaction cost. Here, we present a similar algorithm based on the same idea, that is, repeatedly solving the relaxed problems and gradually approximating the CVaR function with a portion of cutting-plane representation.

To formulate the algorithm, we equivalently rewrite the primal problem by introducing the auxiliary variable u.

$$\min \frac{\lambda}{|S|} \sum_{s \in S} \left(-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v \right) + (1 - \lambda)\, u \tag{4.2a}$$

subject to:

$$u \geq \alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} \left[-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v - \alpha \right]_+ \tag{4.2b}$$

$$u \in \mathbb{R} \tag{4.2c}$$

(4.1b)–(4.1l)

According to the proof in [56], the CVaR constraint (4.2b) is equivalent to the following cutting-plane representation.

$$u \geq \alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in \mathscr{H}} \left(-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v - \alpha \right), \forall \mathscr{H} \subseteq S \tag{4.3}$$

Representation (4.3) obviously contains a series of linear constraints, and the number of constraints is the number of subsets of the scenario set S, i.e., $2^{|S|}$. Since many of the constraints may be redundant, we can simply append the necessary constraints to the relaxed problem iteratively, instead of using all of them directly. The initial relaxed problem of formulation (4.2) is arrived at by replacing the CVaR constraint (4.2b) with constraint (4.4),

$$u \geq U_{min}. \tag{4.4}$$

where U_{min} is a sufficiently small constant to prevent the problem from being unbounded. Accordingly, the feasible region of the initial relaxed problem can be defined by

$$\Omega := \{(\alpha, v, u, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) : \text{(4.1b)–(4.1l), (4.2c), (4.4)}\}, \tag{4.5}$$

where $\boldsymbol{p}$, $\boldsymbol{z}$, and $\boldsymbol{n}$ are the sets of variables p_j, z_i^s, and $n_{j,t}^s$, respectively.

Different from the cutting-plane algorithms in [87], we will not pursue an optimal solution at each iteration, because it takes too much time to execute the branch-and-bound algorithm completely when solving the relaxed MILP problem at each iteration. Instead, we add chosen constraints dynamically to the problem during the branch-and-bound procedure, which is known as *lazy constraint callback.* This callback function is available in optimization software (e.g., Gurobi, CPLEX) and may help to reduce the computing time. With the function, we develop a branch-and-cut algorithm here. The algorithm starts by using the branch-and-bound algorithm to solve the relaxed problem. If a better feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ in Ω is found, it checks whether the corresponding *MIP Gap* is within tolerance. The *MIP Gap* is a relative gap between the upper and lower objective bounds in the branch-and-bound algorithm, which can be obtained directly with optimization software. As long as the *MIP Gap* is unqualified, the callback procedure will be activated. When the feasible solution violates the CVaR constraint (4.2b), cutting planes are generated to separate it from the feasible set. The cut is expressed as:

$$u \geq \alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in \widehat{\mathscr{H}}} \left(-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v - \alpha \right), \tag{4.6}$$

where $\widehat{\mathscr{H}} := \{s \in S : -\sum_{i \in I_s} P_i^s \widehat{z}_i^s + C_1 \sum_{j \in J} \widehat{p}_j + C_2 \widehat{v} - \alpha > 0\}$. **Algorithm 1** is a description of our method.

Algorithm 1 Branch-and-cut Algorithm for Solving Problem (4.2)

Step 1: (Initialization) Let tolerance $\epsilon \geq 0$ for optimality. Define initial feasible region Ω as (4.5)

Step 2: (Branch and bound) Start (or continue) branch-and-bound algorithm to solve the relaxed problem:

$$\min\{\frac{\lambda}{|S|} \sum_{s \in S} (-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v) + (1-\lambda) u : (\alpha, v, u, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) \in \Omega\}.$$

Step 3: (Termination criterion) Once a better feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ in Ω is found, return the solution and current *MIP Gap*. If *MIP Gap* $\leq \epsilon$, stop the algorithm; otherwise go to **Step 4**.

Step 4: (Callback procedure) If the solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ violates the CVaR constraint (4.2b), generate the cutting plane with (4.6) to update the feasible region:

$$\Omega \leftarrow \Omega \cap \{(\alpha, v, u, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) : (4.6)\},$$

and go to **Step 2** to continue branch-and-bound algorithm.

We also prove the convergence of the branch-and-cut algorithm.

Theorem 4.3.1. *For any $\epsilon \geq 0$, the branch-and-cut algorithm has the finite convergence property.*

Proof. Suppose that **Algorithm 1** does not converge in a finite number of iterations. Then *MIP Gap* $> \epsilon$ holds for any sufficiently large number of iterations k such that $k > 2^{|S|}$.

Let us suppose that *MIP Gap* $> \epsilon$ holds at iteration $k > 2^{|S|}$. Since the integer variables are bounded in our model, at **Step 2** of iteration k, we can find either a feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}}) \in \Omega$ that violates the CVaR constraint (4.2b) or an optimal solution of the current model, within a finite number of branch-and-bound iterations. Suppose that we obtain a feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}}) \in \Omega$ violating the CVaR constraint at iteration k. Since we have assumed $k > 2^{|S|}$, this solution satisfies all $k - 1 \geq 2^{|S|}$ inequalities generated before iteration k. Note that the same cutting plane will never be appended twice and the number of cutting planes that can be generated is at most $2^{|S|}$. Thus, the current feasible solution should satisfy all possible cutting planes, which contradicts the assumption that the solution violates the CVaR constraint. On the other hand, if we obtain an optimal solution of the current model, optimality implies that the current *MIP Gap* $\leq \epsilon$ at iteration k, which is a contradiction to the assumption that *MIP Gap* $> \epsilon$ holds. □

4.3.2 Scenario decomposition algorithm

In addition to the cutting-plane representation, we can also transform the nonlinear CVaR function into a linear one based on the lifting representation [76, 87]. Again, the primal problem is equivalently written in the lifting representation as follows.

$$\min \frac{\lambda}{|S|} \sum_{s \in S} \left(- \sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v \right) + (1 - \lambda) \left(\alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} w^s \right) \tag{4.7a}$$

subject to:

$$w^s \geq - \sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v - \alpha \tag{4.7b}$$

$$w^s \in \mathbb{R}_+ \tag{4.7c}$$

$$\text{(4.1b)–(4.1l)}$$

where $w^s, s \in S$ are the auxiliary decision variables. Hence, our formulation becomes a typical two-stage MILP model, which exhibits a block-angular structure and can be exploited in a decomposition fashion. One possible way of solution is to employ the scenario decomposition method, which is based on the Lagrangian decomposition method [36]. Carøe and Schultz [17] first applied scenario decomposition to stochastic integer programming so that the primal problem can be split into more manageable scenario subproblems. The main strategy behind this approach is to create copies of the first-stage variables.

To facilitate subsequent interpretation, we present problem (4.7) in the following compact notation.

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \frac{1}{|S|}\sum_{s\in S} (\boldsymbol{q}^s)^{\mathrm{T}}\, \boldsymbol{y}^s \tag{4.8a}$$

subject to:

$$A\boldsymbol{x} \le \boldsymbol{b} \tag{4.8b}$$

$$T\boldsymbol{x} + W\boldsymbol{y}^s \le \boldsymbol{h}^s, \forall s \in S \tag{4.8c}$$

$$\boldsymbol{x} \in \mathscr{X} \tag{4.8d}$$

$$\boldsymbol{y}^s \in \mathscr{Y}^s, \forall s \in S \tag{4.8e}$$

where $\boldsymbol{c}$, $\boldsymbol{b}$, $\boldsymbol{q}^s$, $s \in S$, and $\boldsymbol{h}^s$, $s \in S$ are the known vectors; A, T, and W are the known matrices; $\boldsymbol{x}$ and $\boldsymbol{y}^s$, $s \in S$ are the first-stage and second-stage decision variable vectors, respectively; and $\boldsymbol{y} = (\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{|S|})$. In our context, it is easy to transform the objective function (4.7a) to (4.8a). The set of constraints (4.8b) is used to represent constraints (4.1b) and (4.1c), while the set of constraints (4.8c) represents constraints (4.1d)–(4.1g) and (4.7b). Finally, the set $\mathscr{X}$ represents constraints (4.1h)–(4.1j), while the set $\mathscr{Y}^s$ denotes constraints (4.1k), (4.1l), and (4.7c).

By replicating the first-stage variables, we can consider the following equivalent formulation.

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|S|}\sum_{s\in S} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^s + (\boldsymbol{q}^s)^{\mathrm{T}}\, \boldsymbol{y}^s\right) \tag{4.9a}$$

subject to:

$$A\boldsymbol{x}^s \le \boldsymbol{b}, \qquad \forall s \in S \tag{4.9b}$$

$$T\boldsymbol{x}^s + W\boldsymbol{y}^s \le \boldsymbol{h}^s, \qquad \forall s \in S \tag{4.9c}$$

$$\boldsymbol{x}^s - \bar{\boldsymbol{x}} = 0, \qquad \forall s \in S \tag{4.9d}$$

$$\boldsymbol{x}^s \in \mathscr{X}, \qquad \forall s \in S \tag{4.9e}$$

$$\boldsymbol{y}^s \in \mathscr{Y}^s, \qquad \forall s \in S \tag{4.9f}$$

$$\bar{\boldsymbol{x}} \in \mathbb{R}^n \tag{4.9g}$$

where $\boldsymbol{x} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^{|S|})$.

Equations (4.9d) are known as the nonanticipativity constraints. These constraints can be represented in various forms [63, 73]. By dualizing the nonanticipativity constraints (4.9d), one may obtain the Lagrangian relaxation of problem (4.10).

$$Z_{LR}(\boldsymbol{\mu}) = \min_{\boldsymbol{x},\boldsymbol{y},\bar{\boldsymbol{x}}} \frac{1}{|S|} \sum_{s\in S} \left(\boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^{\boldsymbol{s}} + (\boldsymbol{q}^{\boldsymbol{s}})^{\mathrm{T}} \boldsymbol{y}^{s} \right) + \sum_{s\in S} (\boldsymbol{\mu}^s)^{\mathrm{T}} (\boldsymbol{x}^{\boldsymbol{s}} - \bar{\boldsymbol{x}}) \tag{4.10}$$

subject to: (4.9b), (4.9c), and (4.9e)–(4.9g).

Since $\bar{\boldsymbol{x}}$ is unconstrained, we bound the Lagrangian with the condition $\sum_{s\in S} \boldsymbol{\mu}^s = 0$ and remove the term $\sum_{s\in S} (\boldsymbol{\mu}^s)^{\mathrm{T}} \bar{\boldsymbol{x}}$ in the Lagrangian function, which makes problem (4.10) separable. It is known that, for any given $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \cdots, \boldsymbol{\mu}^{|S|})$, the Lagrangian relaxation (4.10) provides a valid lower bound on the problem (4.9) [38], i.e., $Z_{LR}(\mu) \leq Z$. To achieve such a bound, we can solve problem (4.10) by dividing it into many simpler subproblems. Note that $Z_{LR}(\mu) = \sum_{s\in S} Z^s_{LR}(\boldsymbol{\mu}^s)$, where $Z^s_{LR}(\boldsymbol{\mu}^s)$ is the objective value of the following subproblem.

$$Z^s_{LR}(\boldsymbol{\mu}^s) = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|S|} \left(\boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^s + (\boldsymbol{q}^s)^{\mathrm{T}} \boldsymbol{y}^s \right) + (\boldsymbol{\mu}^s)^{\mathrm{T}} \boldsymbol{x}^s \tag{4.11}$$

subject to: (4.9b), (4.9c), and (4.9e)–(4.9g), for a given s in S.

To find the best lower bound, we define the Lagrangian dual problem.

$$Z_{LD} = \max_{\boldsymbol{\mu}} \frac{1}{|S|} \sum_{s\in S} Z^s_{LR} \tag{4.12a}$$

subject to:

$$\sum_{s\in S} \boldsymbol{\mu}^s = 0 \tag{4.12b}$$

$$\boldsymbol{\mu}^s \in \mathbb{R}^n, \qquad \forall s \in S \tag{4.12c}$$

This dual problem (4.12) is a concave, nonsmooth optimization problem, which is commonly solved using methods based on subgradients or cutting planes [73]. In this study, a cutting-plane method is used to handle the dual problem. **Algorithm 2** is a stabilized cutting-plane approach with a trust region [44], [54]. Hiriart-Urrut and Lemaréchal [44] proved the convergence of this approach. In this algorithm, the master problem and local problem, respectively, yield an upper bound (Z_{CP}) and lower bound ($Z_{LR}(\mu) = \sum_{s \in S} Z_{LR}^s(\boldsymbol{\mu}^s)$) of Z_{LD}. By iterating the upper and lower bounds in turn, the algorithm terminates when the relative gap between them is less than a given optimality tolerance.

Algorithm 2 Trust-region Cutting-plane Algorithm for Solving Lagrangian Dual Problem (4.12)

```
Step 1: (Initialization) Set the tolerance $\epsilon \geq 0$, iteration count $k = 1$, initial stability
    center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}_k^s = \mathbf{0}$, $s \in S$, ascent coefficient $\omega \in (0,1)$, and initial trust-region $\tau_k \geq 0$,
    solve subproblem (4.11) for all $s$ in $S$, and save $Z_{LR}^s(\boldsymbol{\mu}_k^s)$ and $\boldsymbol{x}_k^s$.
Step 2: (Master problem) Solve the following problem
        $Z_{CP}^k = \max_{\theta,\boldsymbol{\mu}} \sum_{s \in S} \theta^s$
        subject to:
        $\theta^s \leq Z_{LR}^s(\boldsymbol{\mu}_{k'}^s) + (\boldsymbol{x}_{k'}^s)^{\mathrm{T}} (\boldsymbol{\mu}^s - \boldsymbol{\mu}_{k'}^s), \forall s \in S, k' \in \{1, 2, \cdots, k\}$
        $|\boldsymbol{\mu}^s - \bar{\boldsymbol{\mu}}^s| \leq \tau_k, \forall s \in S$
        $\sum_{s \in S} \boldsymbol{\mu}^s = 0$
        $\boldsymbol{\mu}^s \in \mathbb{R}^n, \forall s \in S$
        $\theta^s \in \mathbb{R}, \forall s \in S$
    to obtain $\left(Z_{CP}^k, \boldsymbol{\mu}_{k+1}^s\right)$, and compute
        $\delta_k := Z_{CP}^k - \sum_{s \in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s).$
Step 3: (Termination criterion) If
        $\frac{\delta_k}{1 + \sum_{s \in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s)} \leq \epsilon,$
    stop the algorithm, otherwise go to Step 4.
Step 4: (Local problem) For all $s$ in $S$, solve $|S|$ subproblems (4.11) to obtain the next
    points $\boldsymbol{x}_{k+1}^s$ and $Z_{LR}^s(\boldsymbol{\mu}_{k+1}^s)$.
Step 5: (Center update) If
        $\sum_{s \in S} Z_{LR}^s\left(\boldsymbol{\mu}_{k+1}^s\right) \geq \sum_{s \in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s) + \omega \delta_k,$
    update stability center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}_{k+1}^s, \forall s \in S$, otherwise leave the center unchanged.
Step 6: (Trust region update) Compute the ratio
        $\rho := \frac{\sum_{s \in S} Z_{LR}^s(\boldsymbol{\mu}_{k+1}^s) - \sum_{s \in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s)}{\delta_k}.$
    If $\rho = 1$, then $\tau_{k+1} = 1.5\tau_k$ and if $\rho < 0$, then $\tau_{k+1} = 0.8\tau_k$.
Step 7: (Iteration update) Set $k = k + 1$, go back to Step 2 .
```

Algorithm 2 solves the Lagrangian dual problem (4.12) and derives the lower bound of the primal problem (4.8). Next, we will introduce a simple heuristic to determine the upper bound. The upper bound can be calculated by substituting the feasible first-stage solutions $\boldsymbol{x}^s$ into the primal problem (4.8). In this study, we estimated the feasible first-stage candidates by using the average value of the scenario solution $\boldsymbol{x}^s, s \in S$ and some rounding heuristic to satisfy the integrality restriction ($\boldsymbol{x} = \text{round}(\sum_{s\in S} \boldsymbol{x}^s/|S|)$). When the first-stage variables are known, the primal problem can also be decomposed into $|S|$ subproblems to compute the upper bound of the primal problem (4.8). It is worth noting that if $\boldsymbol{x}$ is not a feasible solution, we can use its upper or lower bound to ensure its feasibility.

Algorithm 3 shows the whole scenario decomposition method combining **Algorithm 2** and the method of computing the upper bound. In detail, it only adds the upper bound update (**Algorithm 3 Step 3**) and a new termination criterion (**Algorithm 3 Step 4**) between **Steps 1** and **2** of **Algorithm 2**.

Algorithm 3 Scenario Decomposition Method for Solving Primal Problem (4.8)

```
Step 1: (Initialization) Set the tolerance $\epsilon_{CP} \geq 0$, $\epsilon_{DG} \geq 0$ iteration count $k = 1$, initial stability center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}^s_k = \mathbf{0}$, $s \in S$, an ascent coefficient $\omega \in (0, 1)$, and initial trust-region $\tau_k \geq 0$. Set upper bound $UB = +\infty$, lower bound $LB = -\infty$
Step 2: (Lower bound update) solve subproblem (4.11) for all $s$ in $S$ to obtain $Z^s_{LR}(\boldsymbol{\mu}^s_k)$ and $\boldsymbol{x}^s_k$; clearly, $Z_{LR}(\mu_k) = \sum_{s\in S} Z^s_{LR}(\boldsymbol{\mu}^s)$. If $Z_{LR}(\boldsymbol{\mu}_k) > LB$, then $LB = Z_{LR}(\boldsymbol{\mu}_k)$.
Step 3: (Upper bound update) Generate the first-stage variable with $\widehat{\boldsymbol{x}}_k = \boldsymbol{x}^s_k/|S|$. For integer variables, $\widehat{\boldsymbol{x}}_k = \text{round}(\boldsymbol{x}^s_k/|S|)$. Then, fix $x = \widehat{\boldsymbol{x}}_k$ in primal problem (4.8) to compute the optimal $Z$. If $Z < UB$, then $UB = Z$.
Step 4: (Termination criterion) If $UB - LB \leq \epsilon_{DG}$, stop the algorithm; otherwise go to Step 5.
Step 5: (Cut generation) Execute Algorithm 2 Step 2 to obtain $\left(Z^k_{CP}, \boldsymbol{\mu}^s_{k+1}, \delta_k\right)$
Step 6: (Termination criterion) If
        $$\frac{\delta_k}{1 + \sum_{s\in S} Z^s_{LR}(\bar{\boldsymbol{\mu}}^s)} \leq \epsilon_{CP},$$
    stop the algorithm, otherwise go to Step 7.
Step 7: (Center and trust region update) Perform Steps 1–3 again to obtain $Z^s_{LR}(\boldsymbol{\mu}^s_k)$, $\boldsymbol{x}^s_k$, $LB$, and $UB$, then update the stability center and trust region with Step 5 and 6 of Algorithm 2.
Step 8: (Iteration update) Set $k = k + 1$, go back to Step 5.
```

Because of the block structure, the scenario decomposition method can split up the primal problem (4.8) into $|S|$ scenario subproblems. In addition, these scenarios can be grouped together into larger blocks to help reduce the gap [17] while maintaining the block structure; thereby, we will rewrite problem (4.9) into problem (4.13).

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|L|} \sum_{l \in L} \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}^l + \frac{1}{|S|} \sum_{l \in L} \sum_{s \in S_l} (\boldsymbol{q}^s)^{\mathrm{T}} \boldsymbol{y}^s \tag{4.13a}$$

subject to:

$$A\boldsymbol{x}^l \leq \boldsymbol{b}, \qquad \forall l \in L \tag{4.13b}$$

$$T\boldsymbol{x}^l + W\boldsymbol{y}^s \leq \boldsymbol{h}^s, \qquad \forall l \in L, s \in S_l \tag{4.13c}$$

$$\boldsymbol{x}^l - \bar{\boldsymbol{x}} = 0, \qquad \forall l \in L \tag{4.13d}$$

$$\boldsymbol{x}^l \in \mathscr{X}, \qquad \forall l \in L \tag{4.13e}$$

$$\boldsymbol{y}^s \in \mathscr{Y}^s, \qquad \forall l \in L, s \in S_l \tag{4.13f}$$

$$\bar{\boldsymbol{x}} \in \mathbb{R}^n \tag{4.13g}$$

where L is the set of blocks and $l \in L$, S_l is the set of scenarios in block l. $|L|$ is chosen to be a suitable divisor of $|S|$. Similarly, we can solve problem (4.13) by the scenario decomposition method. Instead of formulating subproblems for each scenario, we split the scenarios into several large blocks in the order of the scenario, which leads to a better solution.

4.4 Computational experiments

We conducted experiments based on historical data of the Ha:mo RIDE carsharing system in Toyota city, Japan collected from April 1st, 2016 to March 31st, 2017. Ha:mo RIDE is a station-based one-way system. Considering the accessibility of data, we utilized the data from the existing carsharing system directly to generate scenario demand data and assumed the sites of the current stations to be potential location sites. Note that the formulation proposed in Section 4.2 is not affected by these treatments or by the other methods used to obtain the demand data and potential locations (e.g., [9,15]). The model was implemented using Gurobi Optimizer 9.0 in the Python environment on an Intel i7-8700 CPU with 12 cores and 16GB of RAM.

4.4.1 Parameter settings

There were 55 stations with different capacities in the Ha:mo RIDE system. As mentioned above, the sites of these stations were assumed to be the potential location sites and the current station capacities were regarded as maximum numbers of parking spaces that can be set. The operating time of the system was from 6:00 to 24:00, which in the experiments was divided into multiple 5-minute intervals.

An important part of the parameters in the proposed model is the scenario demand data. Instead of predicting the potential demand with forecast methods like regression prediction, we generated the demand by using a Poisson distribution due to the limited data, which is the same as [95]. The average OD matrices for the same hours were calculated from the historical data for different days. The number of OD trips per hour was then evenly distributed over the time intervals, which determines the parameter in the Poisson distribution. In such a way, we generated a random number of possible trips departing from each station for each time interval. Furthermore, the arrival time was determined by adding the trip duration to the departure time. The trip duration for each OD pair varied every hour, which was obtained from the Google Maps Distance Matrix API. We based the trip price on the trip duration: a fare of ¥200 for up to 10 minutes and ¥20 per minute after that. Using this procedure, we repeatedly generated different scenario demands.

The remaining parameters are mostly related to the costs. Besides the station cost, other costs are constant ones, including the station and vehicle operating costs and the cost of purchasing vehicles. Because the station cost involves the land cost, which varies according to the local conditions, we collected the land cost data for Toyota City from the National Land Price Map, Japan [75]. The constant parameters are summarized in Table 1. Note that some parameters have been modified to ensure the system can make a profit.

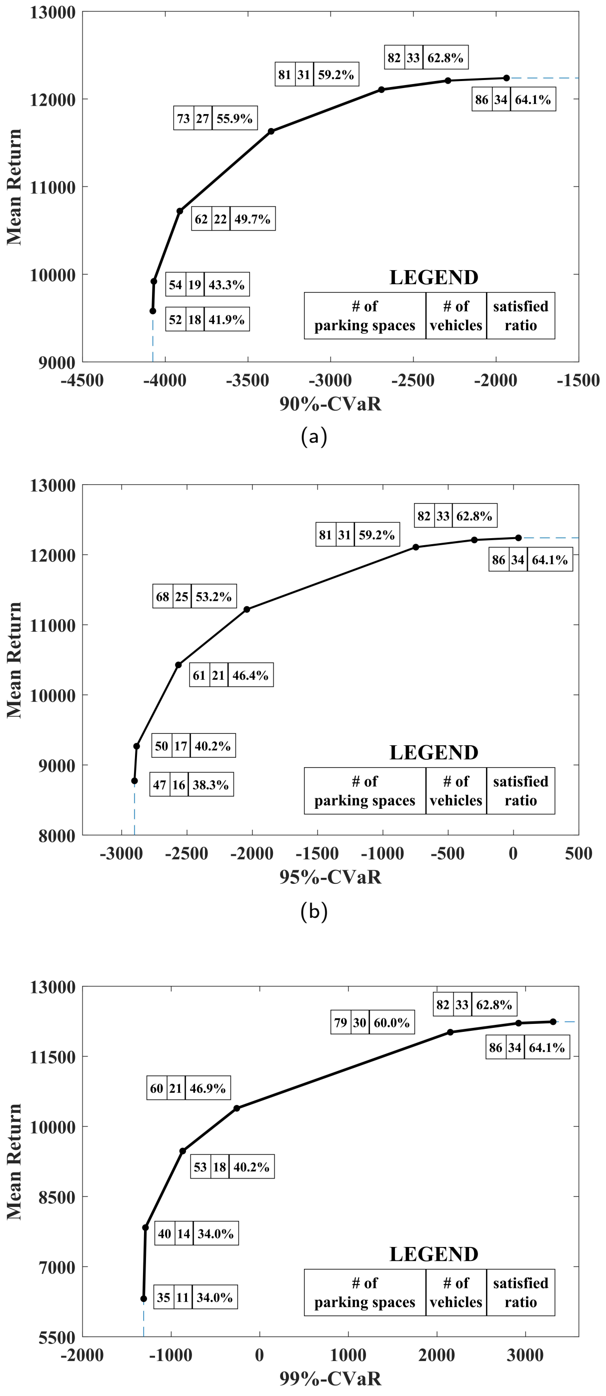
Table 4.1: Values of constant parameters used in the model.

Parameters	Value
Operating cost per parking space per scenario (C_1)	¥100
Operating cost per vehicle per scenario(C_2)	¥200
Cost for purchasing a vehicle (F)	¥879000
Available budget (B)	¥200 million
Number of scenarios ($\|S\|$)	1000

4.4.2 Optimization results

With the given parameters, we solved the strategic decision model directly with Gurobi and generated efficient frontiers for the 90%, 95%, and 99% confidence levels by using different weight values, as shown in Figure 4.1. This figure also shows optimal results including the total number of parking spaces, the number of vehicles used, and the demand satisfaction rate for different weight values. From the efficient frontiers, it is obvious that high returns are accompanied by high risks. To achieve higher returns, the main approach appears to be increasing the number of parking spaces or vehicles, which improves the satisfied demand ratio. Observing the changes of the axes in the figures, we obtained the general trend of these frontiers where a higher confidence level quantifies more serious risk and reduces the return to a certain degree. Furthermore, it seems sufficient to set 86 parking spaces and purchase 34 vehicles in the carsharing system, since more parking spaces or vehicles increases risk, not return.

Figure 4.2 illustrates the optimal location and capacity of stations for different values of λ, given $\beta = 95\%$. The size of the circle represents the number of parking spaces. The individual figures show the optimal number of stations, number of parking spaces, and required fleet size in the system at the upper right corner. Generally, stations equipped with more parking spaces are mainly located at the spots with high demand, such as at Toyota factories and railway stations. However, these hot spots are easily affected by the weight values; that is, the number of parking spaces will increase when more attention is paid to the return. In comparison, the solutions seem more robust for some small stations. There are 13 common small stations, marked in red circles, where the location and capacity of stations are the same.

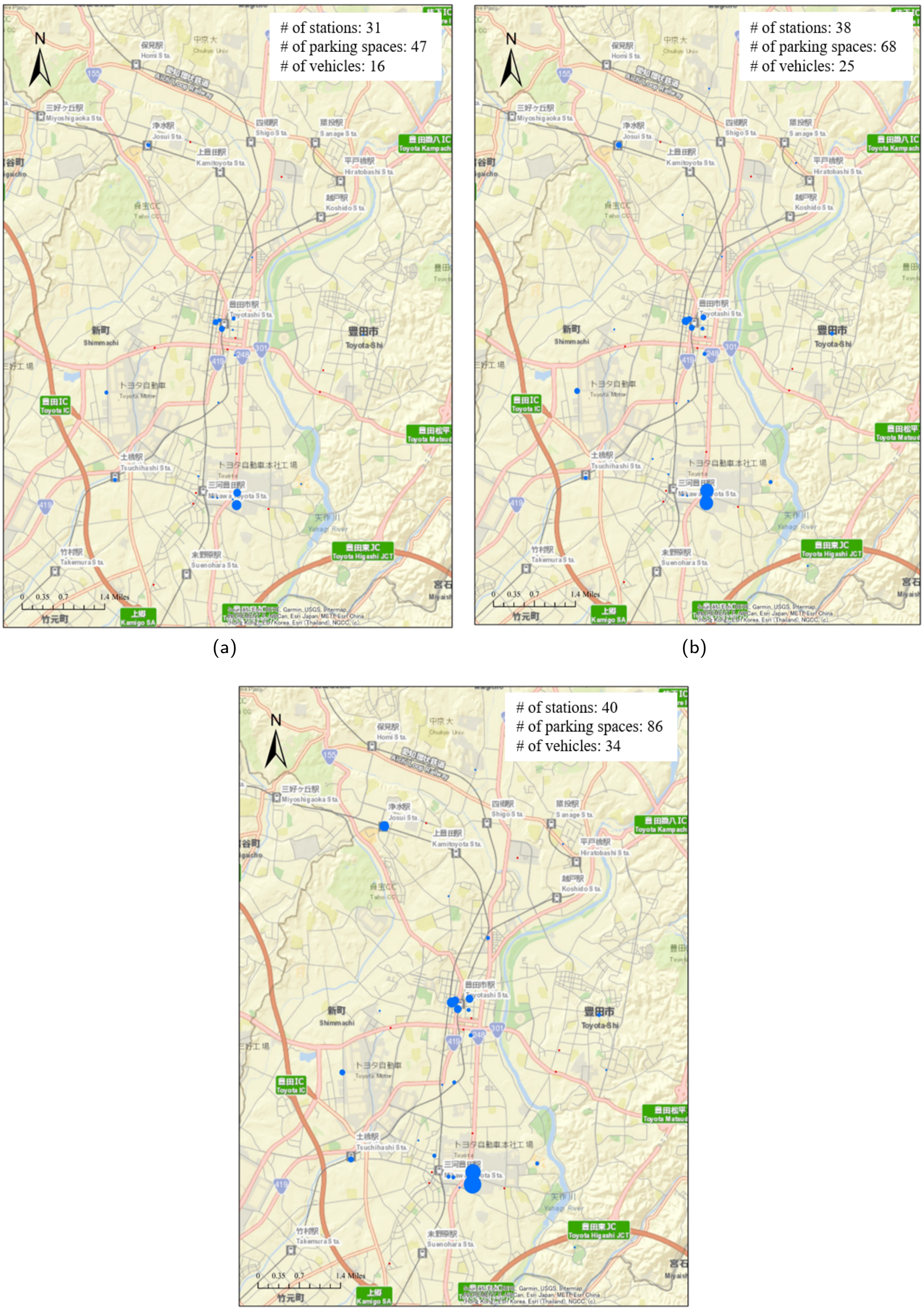


(a)

(b)

(c)

Figure 4.1: Efficient frontiers of mean return and CVaR: (a) $\beta = 90\%$; (b) $\beta = 95\%$; (c) $\beta = 99\%$.



(a)

(b)

(c)

Figure 4.2: Optimal station locations and capacities for $\beta = 95\%$: (a) $\lambda = 0$; (b) $\lambda = 0.5$; (c) $\lambda = 1$.

4.4.3 Out-of-sample performance of the risk-averse model

As mentioned above, most of the previous studies maximized the expectation value of returns in the optimization model without any consideration of risk. In this subsection, we develop an evaluation method that is similar to the training and testing method used in the field of machine learning to verify whether introducing risk has benefits, i.e., whether the strategic decisions from risk-averse model are better. As can be seen in Figure 4.3, the proposed evaluation process consists of two important parts. In the first part (training part), we set different weight values in the strategic model to optimize the corresponding strategic decisions. In the second part (testing part), we render these decisions as additional constraints in the strategic model and meanwhile input test demand data into the model. After optimizing other variables according to the determined variables associated with the strategic decisions, the mean returns in the objective function are compared to further inspect the strategic decisions.

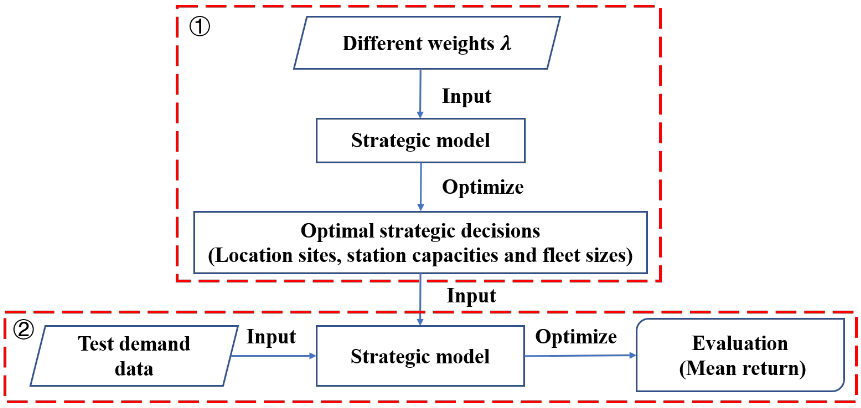


Figure 4.3: Scheme to evaluate the model with risk term.

To examine the impact of introducing the risk term on the strategic decisions, the weight values (λ) used to generate the efficient frontier are divided into three categories: i) $\lambda = 0$, i.e., only risk is considered; ii) $0 < \lambda < 1$, i.e., both return and risk are considered; iii) $\lambda = 1$, i.e., only return is considered. Given different confidence levels β, the strategic decisions with respect to different λ will be evaluated by the indicator, mean return, on diverse test demand data. We generated 10 sets of test demand data that followed a Poisson distribution. The test results for $\beta = 90\%, 95\%$, and 99% are reported in Tables 4.2, 4.3, and 4.4, respectively.

For each set of test data, the maximal return is marked in bold. In Table 4.2, it is interesting that, for most of the test data, the maximal returns are obtained when $0 < \lambda < 1$ rather than $\lambda = 1$. Similar observations apply to Tables 4.3 and 4.4. Strategic decisions based on a single criterion are more likely to cause poor performance under demand uncertainty, which indicates the necessity of introducing the risk term. Weighting the return against risk, the carsharing system operator may earn higher returns in the future. In addition, we also find that appropriate attention should be paid to risk (i.e., choosing a suitably smaller λ) if the demand data results in lower returns. Take test data 1 and 10 in Table 2 as examples. The maximal returns are obtained when $\lambda = 0.1$ for test data 10 and $\lambda = 0.999$ or 1 for test data 1, and it can be seen that the return from test data 10 is lower than the return from test data 1.

Table 4.2: Mean return on test data for different λ when $\beta = 90\%$.

Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	5335	9506	9830	10628	11485	11920	11990	**12022**	**12022**
2	4140	8593	8853	9511	10135	10388	**10397**	10340	10340
3	3980	8296	8514	9033	9566	**9790**	9734	9708	9708
4	3190	7956	8210	8681	9122	**9217**	9154	9138	9138
5	3191	7530	7759	8236	8524	**8567**	8490	8350	8350
6	3065	6845	7011	7331	**7394**	7328	7142	7019	7019
7	3139	6692	6794	7038	**7155**	6944	6720	6583	6583
8	3589	6396	6494	6670	**6684**	6476	6238	6078	6078
9	2659	5399	5460	**5532**	5178	4709	4403	4207	4207
10	2792	4411	**4461**	4359	3815	3161	2764	2459	2459
Average	3508	7162	7339	7702	**7906**	7850	7703	7590	7590

Table 4.3: Mean return on test data for different λ when $\beta = 95\%$.

Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	3881	8762	9200	10310	11088	11920	11990	**12022**	**12022**
2	2889	7887	8336	9303	9890	10388	**10397**	10340	10340
3	3604	7632	8048	8795	9322	**9790**	9734	9708	9708
4	2655	7365	7735	8509	8938	**9217**	9154	9138	9138
5	2759	6871	7313	8026	8408	**8567**	8490	8350	8350
6	2790	6272	6646	7213	**7377**	7328	7142	7019	7019
7	2727	6172	6531	6971	**7079**	6944	6720	6583	6583
8	2805	5913	6241	6630	**6660**	6476	6238	6078	6078
9	2598	5071	5321	**5537**	5376	4709	4403	4207	4207
10	2215	4163	**4375**	4374	4063	3161	2764	2459	2459
Average	2892	6611	6975	7567	7820	**7850**	7703	7590	7590

Table 4.4: Mean return on test data for different λ when $\beta = 99\%$.

Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	1757	6254	7755	9391	10289	11849	11990	**12022**	**12022**
2	1937	5702	7041	8519	9206	10330	**10397**	10340	10340
3	2844	5569	6919	8166	8806	**9766**	9734	9708	9708
4	1747	5372	6656	7875	8438	**9220**	9154	9138	9138
5	1754	5065	6144	7476	7985	**8542**	8490	8350	8350
6	1033	4716	5716	6791	7218	**7334**	7142	7019	7019
7	388	4660	5574	6537	6962	**7010**	6720	6583	6583
8	1738	4441	5341	6242	**6589**	6474	6238	6078	6078
9	1036	3892	4547	5345	**5457**	4776	4403	4207	4207
10	1612	3343	3785	**4384**	4309	3263	2764	2459	2459
Average	1585	4901	5948	7073	7526	**7856**	7703	7590	7590

4.4.4 Comparison of proposed solution methods

To assess the efficiency of our algorithms, we conducted computational experiments comparing the two proposed algorithms with direct usage of the Gurobi optimization solver and Benders decomposition-based algorithm used in [62]. For each parameter set (λ, β, and $|S|$), we designed three experiments by generating different trip demands in each scenario, and we evaluated the average performance of these experiments. We set the weight parameter, $\lambda \in \{0.1, 0.5, 0.9\}$, the confidence level parameter, $\beta \in \{90\%, 95\%, 99\%\}$, and the number of scenarios, $|S| \in \{200, 1000, 2000\}$. When using Gurobi directly and employing the branch-and-bound algorithm, both optimality tolerances take the default value, $\epsilon = 10^{-4}$. In the scenario decomposition algorithm, the optimality tolerances ϵ_{CP} and ϵ_{DG} are 10^{-4} and 300, respectively. For all methods, the time limits are set to be 7200s.

In Tables 4.5, 4.6, and 4.7, "Direct" means direct usage of Gurobi, "B&C" is the branch-and-cut algorithm, "SD" represents the scenario decomposition method, and "BD" means the Benders decomposition-based algorithm introduced in [62]. On large-scale problems, the algorithms usually terminated because they ran out of memory. Accordingly, "OM(*)" means that the algorithm experienced a memory shortage, where "*" is the number of memory shortages out of three. Similarly, "OT(*)" depicts the computation was terminated due to time limit, where "*" is the number out of three. Moreover, "Gap (%)" is the relative optimality gap.

Tables 4.5, 4.6, and 4.7 illustrate the results of solving the problems with different numbers of scenarios, mainly including the gaps and time. As can be seen in Tables 4.5 and 4.6, both direct method and B&C algorithm can provide optimal values with gaps being less than 10^{-4}. For some problems with 200 scenarios, the B&C algorithm takes a little more time than the direct approach. However, when the number of scenarios is 1000, the B&C algorithm shows more favorable results; its computing time was reduced by 51% on average in comparison with Gurobi. Turning now to the SD algorithm, we see that the algorithm is faster than both the direct approach and the B&C algorithm and the resulting optimality gaps are mostly within 3%. Although the SD algorithm can provide tight lower bounds with the trust-region cutting-plane method, a simple heuristic to estimate the upper bounds sometimes may cause weak solutions and large gaps (e.g., $\lambda = 0.1$, $\beta = 99\%$), which can be further improved. In comparison, BD performs worse than all other methods. For all problems with 200 scenarios or 1000 scenarios, we did not obtain good solutions within the specified time by using the BD algorithm. With the increase of the number of iterations, the master problem in the BD algorithm becomes a larger MILP gradually and most of the time is spent on solving the master problem. However, in the SD algorithm, we substituted a heuristic solution (average value) into the primal problem to get the solution without solving such an MILP. In short, both B&C algorithm and SD algorithm are effective when solving the problem with fewer scenarios, and one can choose the B&C algorithm and SD algorithm according to the need for a smaller gap or for higher speed .

Now, let us look at Table 4.7, which indicates the advantage and efficiency of the SD algorithm for solving large-scale problem. The core of the B&C algorithm is replacing the CVaR constraint with the cutting-plane representation, so a problem remains is that the other linear constraints still depend on the number of scenarios, i.e., a much greater number of scenarios may result in difficulty finding a good solution. The table indicates that the direct approach and B&C algorithm could not solve the problems with 2000 scenarios and all experiments terminated due to memory shortage. BD method could not find a feasible solution for our model within the given time. Nevertheless, the SD algorithm can compensate for hardware defects by consuming more (but still tolerable) computation time and obtain favorable solutions. Even for the problems with $\lambda = 0.1$ and $\beta = 99\%$, the gap presents a downward trend compared with the values in Tables 4.5 and 4.6. These illustrate the superiority of SD on large-scale problems.

Table 4.5: Results of solving problem with 200 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
0.1	90%	≤ 0.01		5.33	> 100	58.0	64.7	33.2	OT(3)
	95%			12.35		46.3	71.2	40.3	
	99%			24.62		150.0	231.0	88.7	
0.5	90%			1.73		59.1	50.8	33.3	
	95%			2.60		68.5	50.8	29.6	
	99%			3.28		82.5	86.5	36.9	
0.9	90%			0.35		64.8	37.1	23.4	
	95%			0.46		57.3	43.2	23.5	
	99%			0.81		47.5	50.9	21.0	

Table 4.6: Results of solving problem with 1000 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
0.1	90%	≤ 0.01		1.08	>100	3587.3	2312.3	1132.0	OT(3)
	95%			2.48		4035.3	2884.4	1025.0	
	99%			17.16		4137.8	2966.4	1430.4	
0.5	90%			0.29		5202.5	2057.7	1147.6	
	95%			0.97		3595.5	1597.8	1063.3	
	99%			2.91		3232.4	1958.3	1059.4	
0.9	90%			0.06		3044.1	907.5	949.5	
	95%			0.06		3582.9	1003.1	878.4	
	99%			0.22		3299.2	1144.2	867.1	

Table 4.7: Results of solving problem with 2000 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
0.1	90%	-		1.94	-	OM(3)		1804.6	OT(3)
	95%			5.57				OT(3)	
	99%			13.96				4216.5	
0.5	90%			0.60				1924.4	
	95%			1.14				2243.5	
	99%			4.52				OT(3)	
0.9	90%			0.20				3739.5	
	95%			0.21				2957.9	
	99%			0.45				6666.1	

4.5 Summary

We proposed a two-stage stochastic risk-averse MINLP model to optimize the strategic decisions in one-way station-based carsharing systems operating under demand uncertainty. In the model, the optimal location, capacity of stations, and fleet size can be determined at the same time. In addition to the expected return that is a common optimization objective, the risk measure CVaR is incorporated into the model so that the operator can examine the trade-off between return and risk. Since we aimed to solve the problem efficiently, we developed two methods, a branch-and-cut algorithm and a scenario decomposition algorithm, by converting the primal problem into two different equivalent problems.

Using the historical demand data, we generated scenario demands that followed a Poisson distribution and conducted computational experiments. By solving the problems with different weight values, we obtained efficient frontiers, which revealed a positive correlation between return and CVaR. From the efficient frontiers, we found that building more parking spaces or preparing more vehicles can improve the return and the satisfied demand ratio, but that they cause more risk. The stations equipped with more parking spaces were suggested to be set at the spots with high demand, such as at Toyota factories and railway stations. Additionally, we evaluated the advantages of the proposed model with the risk term. The results show that it is better to consider both the return and risk in the objective function so that the strategic decisions may lead to a higher return. When comparing the solution methods, the algorithm developed by Lu et al. [62] has unsatisfactory performance on our model. By contrast, both branch-and-cut and scenario decomposition algorithms are effective at solving small- and medium-scale problems. More importantly, the scenario decomposition can deal with large-scale problems that cannot be solved by the direct approach or by the branch-and-cut algorithm.

Chapter 5

Conclusions and outlook

5.1 Conclusions

The thesis focused on the strategic design for two types of emerging mobility systems: bus transit systems operating with human-driven and autonomous buses and one-way station-based carsharing systems, which correspond to two gradual transformations in today's mobilities. Mathematical formulations and approaches were applied to determine the various planning decisions in each type of system.

In the emerging transit systems, we developed an MINLP (Model I)and its approximately linear form to solve the BDL&TNDP with limited drivers, where the bus depot locations, fleet size, bus routes, and frequency and bus type of each route were optimized jointly. Given the complexity of linearized Model I, a simplified path-based model (Model II) and its linear approximation were introduced. Compared with Model I, Model II was more computationally efficient due to fewer variables and constraints. To assess the solutions obtained from Model II, we substituted the solutions into Model I and compared the objective values of the new Model I with those of the original Model I. By using the networks from previous studies, two linearized models were tested with the given parameters. We first generated the efficient frontiers of operators' and users' costs by varying the weights and analyzed the effect of the number of available drivers. From the results, we observed that the users' costs tend to be high if no autonomous buses are introduced with the limitation of drivers. Introducing autonomous buses helps construct a user-centric transit system. Then, we analyzed the model size and computational efficiency for both models. In comparison with Model I, Model II can reach the optimality gap quickly because of the smaller model size. In the last part, the differences between the new Model I and the original Model I were computed. We found that different candidate paths of passengers and coefficient waiting time will affect the difference. Generally, more candidate travel paths of passengers will lead to a smaller difference. Model II can obtain favorable solutions within a few seconds on two test networks.

For one-way station-based carsharing systems, we proposed a mean-CVaR model to optimize the strategic decisions considering demand uncertainty. In the model, the optimal location, capacity of stations, and fleet size can be determined simultaneously. We developed a branch-and-cut algorithm and a scenario decomposition algorithm to solve the problem efficiently by converting the original problem into two different equivalent problems. With the generated scenario demands, we conducted computational experiments to solve the problems with different weight values so that the efficient frontiers were obtained, which revealed a positive correlation between return and CVaR. The results show that building more parking spaces or preparing more vehicles can improve the return and the satisfied demand ratio, but they cause much risk. The stations equipped with more parking spaces are usually set at the spots with high demand, such as at Toyota factories and railway stations. Besides, we analyzed the proposed model with the risk term by using a training and testing method. We found it is better to consider both the return and risk in the objective function so that the strategic decisions may cause a higher return when demand data is uncertain. In addition, compared with the algorithm in [62], both branch-and-cut and scenario decomposition algorithms are effective at solving small- and medium-scale problems. More importantly, scenario decomposition can handle large-scale problems that other methods cannot solve.

5.2 Outlook

As for the emerging bus transit systems, we made several simplifications when solving the BDL&TNDP. Some improvements can be taken into account in future work. We ignored the bus capacity considering the optimal strategies assignment model is unsuitable for the congested area. Based on this relaxation, the proposed model can be improved by integrating the congested transit assignment model (e.g., [18]). In this study, we applied the autonomous buses to fix-route design. If the autonomous buses can travel on flexible routes in light of the traffic condition, both operators' and users' costs may be reduced.

Concerning the carsharing systems, an interesting challenge would be to develop more complicated variations of our model with constraints such as energy consumption, vehicle relocations, or relays, which will relax our assumptions. In particular, given the current scale of the proposed model, relaxation tricks might be needed to handle these constraints. On the other hand, from a computational perspective, we can improve the heuristic method for determining upper bounds in scenario decomposition to make the algorithm more sophisticated.

In addition, we designed the two emerging systems separately in the thesis. As the concept of MaaS becomes widespread, there is a trend to integrate different modes of transportation. Hence, it is also a crucial issue to design integrated multimodal transportation from different operators' perspectives.

Bibliography

[1] Fatima A. Ali and Noha M. Hassan. Optimization of bus depot location with consideration of maintenance center availability. *Journal of Transportation Engineering, Part A: Systems*, 144(2):05017011, 2018.

[2] Mohamed H. Baaj and Hani S. Mahmassani. An AI-based approach for transit route system planning and design. *Journal of Advanced Transportation*, 25(2):187–209, 1991.

[3] Mohamed H. Baaj and Hani S. Mahmassani. Hybrid route generation heuristic algorithm for the design of transit networks. *Transportation Research Part C: Emerging Technologies*, 3(1):31–50, 1995.

[4] Saeed A. Bagloee and Avishai A. Ceder. Transit-network design methodology for actual-size road networks. *Transportation Research Part B: Methodological*, 45(10):1787–1804, 2011.

[5] Matthew Barth and Susan A. Shaheen. Shared-use vehicle systems: Framework for classifying carsharing, station cars, and combined approaches. *Transportation Research Record: Journal of the Transportation Research Board*, 1791:105–112, 1 2002.

[6] Oscar Bergqvist and Mikaela Åstrand. *Bus Line Optimisation Using Autonomous Minibuses*. Bachelor's thesis, KTH Royal Institute of Technology, 2017.

[7] Ralf Borndörfer, Martin Grötschel, and Marc E. Pfetsch. *A Path-based Model for Line Planning in Public Transport*. ZIB, 2005.

[8] Jo Borrás. New Flyer Xcelsior AV is America's first autonomous bus. `https://cleantechnica.com/2021/02/02/new-flyer-xcelsior-av-is-americas-first-autonomous-bus/`, Feb 2021. [Online; accessed 1-September-2021].

[9] Burak Boyacı, Konstantinos G. Zografos, and Nikolas Geroliminis. An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, 240(3):718–733, 2015.

[10] Burak Boyacı, Konstantinos G. Zografos, and Nikolas Geroliminis. An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B: Methodological*, 95:214–237, 2017.

[11] Georg Brandstätter, Claudio Gambella, Markus Leitner, Enrico Malaguti, Filippo Masini, Jakob Puchinger, Mario Ruthmair, and Daniele Vigo. Overview of optimization problems in electric car-sharing system design and management. In *Dynamic Perspectives on Managerial Decision Making*, pages 441–471. Springer, 2016.

[12] Georg Brandstätter, Michael Kahr, and Markus Leitner. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. *Transportation Research Part B: Methodological*, 104:17–35, 2017.

[13] Maurizio Bruglieri, Alberto Colorni, and Alessandro Luè. The relocation problem for the one-way electric vehicle sharing. *Networks*, 64(4):292–305, 2014.

[14] Michael R. Bussieck, Peter Kreuzer, and Uwe T. Zimmermann. Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63, 1997.

[15] Hatice Çalık and Bernard Fortz. A benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty. *Transportation Research Part B: Methodological*, 125:121–150, 2019.

[16] Héctor Cancela, Antonio Mauttone, and María E. Urquhart. Mathematical programming formulations for transit network design. *Transportation Research Part B: Methodological*, 77:17–37, 2015.

[17] Claus C. Carøe and Rüdiger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

[18] Avishai Ceder and Nigel H. M. Wilson. Bus network design. *Transportation Research Part B: Methodological*, 20(4):331–344, 1986.

[19] Chao Chen, Baozhen Yao, Gang Chen, and Zhihui Tian. A queuing–location–allocation model for designing a capacitated bus garage system. *Engineering Optimization*, pages 1–18, 2021.

[20] Zhihao Chen. *Strategic Network Planning under Uncertainty with Two-Stage Stochastic Integer Programming.* PhD thesis, University of Michigan, 2016.

[21] Kenichiro Chinen, Yang Sun, Mitsutaka Matsumoto, and Yoon-Young Chun. Towards a sustainable society through emerging mobility services: A case of autonomous buses. *Sustainability*, 12(21):9170, 2020.

[22] Roberto Cominetti and José Correa. Common-lines and passenger assignment in congested transit networks. *Transportation Science*, 35(3):250–267, 2001.

[23] Scott Corwin, Joe Vitale, Eamonn Kelly, and Elizabeth Cathles. The future of mobility: How transportation technology and social trends are creating a new business ecosystem. `https://www2.deloitte.com/us/en/insights/focus/future-of-mobility/transportation-technology.html`, 2015. [Online; accessed 20-August-2021].

[24] Mark S. Daskin. *Network and Discrete location: Models, Algorithms, and Applications.* John Wiley & Sons, 2011.

[25] Gonçalo H. de Almeida Correia and António P. Antunes. Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):233–247, 2012.

[26] Gonçalo H. de Almeida Correia, Diana R. Jorge, and David M. Antunes. The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal. *Journal of Intelligent Transportation Systems*, 18(3):299–308, 2014.

[27] Joaquín de Cea and Enrique Fernández. Transit assignment for congested public transport systems: An equilibrium model. *Transportation Science*, 27(2):133–147, 1993.

[28] Alicia De-Los-Santos, David Canca, and Eva Barrena. Mathematical formulations for the bimodal bus-pedestrian social welfare network design problem. *Transportation Research Part B: Methodological*, 145:302–323, 2021.

[29] Guy Desaulniers and Mark D. Hickman. Public transit. *Handbooks in Operations Research and Management Science*, 14:69–127, 2007.

[30] Zvi Drezner and Horst W. Hamacher. *Facility Location: Applications and Theory.* Springer Science & Business Media, 2001.

[31] Javier Duran, Lorena Pradenas, and Victor Parada. Transit network design with pollution minimization. *Public Transport*, 11(1):189–210, 2019.

[32] Javier Durán-Micco. *Transit Network Design: Narrowing the Gap between Theory and Practice.* PhD thesis, Katholieke Universiteit Leuven, 2021.

[33] Lang Fan and Christine L Mumford. A metaheuristic approach to the urban transit routing problem. *Journal of Heuristics*, 16(3):353–372, 2010.

[34] Wei Fan. Management of dynamic vehicle allocation for carsharing systems: Stochastic programming approach. *Transportation Research Record*, 2359(1):51–58, 2013.

[35] Francesco Ferrero, Guido Perboli, Mariangela Rosano, and Andrea Vesco. Car-sharing services: An annotated review. *Sustainable Cities and Society*, 37:501–518, 2018.

[36] Marshall L. Fisher. An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15(2):10–21, 1985.

[37] Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. Design and management of vehicle-sharing systems: a survey of algorithmic approaches. *Smart Cities and Homes*, pages 261–289, 2016.

[38] Arthur M. Geoffrion. Lagrangean relaxation for integer programming. In *Approaches to Integer Programming*, pages 82–114. Springer, 1974.

[39] Bruce Golden, Arjang Assad, Larry Levy, and Filip Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

[40] Jan-Willem Goossens, Stan van Hoesel, and Leo Kroon. A branch-anc-cut approach for solving line planning problems. *Transportation Science*, 38(3):379–393, 2004.

[41] Aric Hagberg, Pieter Swart, and Daniel S. Chult. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[42] Jonas Hatzenbühler, Oded Cats, and Erik Jenelius. Network design for line-based autonomous bus services. *Transportation*, pages 1–36, 2021.

[43] Shuimiao He, Yuanqing Wang, and Zihe Zhang. Optimization of bus depot location with consideration of dead kilometers: A case study in Xi'an, China. In *CICTP 2019: Transportation in China–Connecting the World*, pages 4987–4997, 2019.

[44] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms II – Advanced Theory and Bundle Methods*, volume 305. Springer science & business media, 2013.

[45] Kai Huang, Kun An, Jeppe Rich, and Wanjing Ma. Vehicle relocation in one-way station-based electric carsharing systems: A comparative study of operator-based and user-based methods. *Transportation Research Part E: Logistics and Transportation Review*, 142:102081, 2020.

[46] Kai Huang, Gonçalo H. de Almeida Correia, and Kun An. Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand. *Transportation Research Part C: Emerging Technologies*, 90:1–17, 2018.

[47] Omar J. Ibarra-Rojas, Felipe Delgado, Ricardo Giesen, and Juan Carlos Muñoz. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, 77:38–75, 2015.

[48] Oliver Ibe. *Fundamentals of Applied Probability and Random Processes.* Academic Press, 2014.

[49] Christina Iliopoulou, Konstantinos Kepaptsoglou, and Eleni Vlahogianni. Metaheuristics for the transit route network design problem: A review and comparative analysis. *Public Transport*, 11(3):487–521, 2019.

[50] Stefan Illgen and Michael Höck. Literature review of the vehicle relocation problem in one-way car sharing networks. *Transportation Research Part B: Methodological*, 120:193–204, 2019.

[51] Diana Jorge, Gonçalo H. de Almeida Correia, and Cynthia Barnhart. Testing the validity of the MIP approach for locating carsharing stations in one-way systems. *Procedia-Social and Behavioral Sciences*, 54:138–148, 2012.

[52] Diana Jorge, Gonçalo H. de Almeida Correia, and Cynthia Barnhart. Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1667–1675, 2014.

[53] Richard Katzev. Car sharing: A new approach to urban transportation problems. *Analyses of Social Issues and Public Policy*, 3(1):65–86, 2003.

[54] Brage R. Knudsen, Ignacio E. Grossmann, Bjarne Foss, and Andrew R. Conn. Lagrangian relaxation based decomposition for well scheduling in shale-gas systems. *Computers & Chemical Engineering*, 63:234–249, 2014.

[55] Pavlo Krokhmal, Jonas Palmquist, and Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, 4:43–68, 2002.

[56] Alexandra Künzi-Bay and János Mayer. Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, 3(1):3–27, 2006.

[57] Fumitaka Kurauchi, Michael G.H. Bell, and Jan-Dirk Schmöcker. Capacity constrained transit assignment with common lines. *Journal of Mathematical Modelling and Algorithms*, 2(4):309–327, 2003.

[58] Richard C. Larson and Amedeo R. Odoni. *Urban Operations Research.* Monograph, 1981.

[59] Helene Lidestam, Carolina Camén, and Björn Lidestam. Evaluation of cost drivers within public bus transports in sweden. *Research in Transportation Economics*, 69:157–164, 2018.

[60] Todd Litman. Autonomous vehicle implementation predictions: Implications for transport planning. `https://nationalcenterformobilitymanagement.org/wp-content/uploads/2020/03/avip.pdf`, Jan 2020. [Online; accessed 5-September-2021].

[61] Yi Liu, Xuesong Feng, Lukai Zhang, Weixing Hua, and Kemeng Li. A pareto artificial fish swarm algorithm for solving a multi-objective electric transit network design problem. *Transportmetrica A: Transport Science*, 16(3):1648–1670, 2020.

[62] Mengshi Lu, Zhihao Chen, and Siqian Shen. Optimizing the profitability and quality of service in carshare systems under demand uncertainty. *Manufacturing & Service Operations Management*, 20(2):162–180, 2018.

[63] Miles Lubin, Kipp Martin, Cosmin G. Petra, and Burhaneddin Sandıkçı. On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters*, 41(3):252–258, 2013.

[64] Thomas L. Magnanti and Richard T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[65] Thomas H. Maze, Snehamay Khasnabis, and Mehmet D. Kutsal. Optimization methodology for bus garage locations. *Transportation Engineering Journal of ASCE*, 108(6):550–569, 1982.

[66] Thomas H. Maze, Snehamay Khasnabis, and Mehmet D. Kutsal. Application of a bus garage location and sizing optimization. *Transportation Research Part A: General*, 17(1):65–72, 1983.

[67] Jyotiprasad Medhi. *Stochastic Models in Queueing Theory*. Academic Press, 2002.

[68] Leszek Mindur, Grzegorz Sierpiński, and Katarzyna Turoń. Car-sharing development–current state and perspective. *Logistics and Transport*, 39(3):5–14, 2018.

[69] Christine L. Mumford. New heuristic and evolutionary operators for the multi-objective urban transit routing problem. In *2013 IEEE Congress on Evolutionary Computation*, pages 939–946. IEEE, 2013.

[70] Rahul Nair and Elise Miller-Hooks. Fleet management for vehicle sharing operations. *Transportation Science*, 45(4):524–540, 2011.

[71] Kyodo News. Softbank unit to begin Japan's 1st public road autonomous bus service. `https://english.kyodonews.net/news/2020/11/90a5ee06e959-softbank-unit-to-begin-japans-1st-public-road-autonomous-bus-service.html`, Nov 2020. [Online; accessed 1-September-2021].

[72] Nilay Noyan. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559, 2012.

[73] Fabricio Oliveira, Vijay Gupta, Silvio Hamacher, and Ignacio E. Grossmann. A lagrangean decomposition approach for oil supply chain investment planning under uncertainty with risk considerations. *Computers & Chemical Engineering*, 50:184–195, 2013.

[74] Georg Ch Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In *Probabilistic Constrained Optimization*, pages 272–281. Springer, 2000.

[75] Research Center for Property Assessment System. National land price map. `https://www.chikamap.jp/chikamap/Portal?mid=216`, 2019. [Online; accessed 26-July-2019].

[76] Ralph T. Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[77] Piotr Sawicki and Szymon Fierek. Mixed public transport lines construction and vehicle's depots location problems. In *Scientific And Technical Conference Transport Systems Theory And Practice*, pages 213–224. Springer, 2017.

[78] Anita Schöbel. Line planning in public transportation: Models and methods. *OR Spectrum*, 34(3):491–510, 2012.

[79] Anita Schöbel and Susanne Scholl. Line planning with minimal traveling time. In *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.

[80] Robert Sedgewick. *Algorithms in C, Part 5: Graph Algorithms.* Pearson Education, 2001.

[81] Sergej S. Shadrin and Anastasiia A. Ivanova. Analytical review of standard SAE J3016 «Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles» with latest updates. *Avtomobil'. Doroga. Infrastruktura.*, 3(21):10, 2019.

[82] Susan A. Shaheen, Nelson D. Chan, and Helen Micheaux. One-way carsharing's evolution and operator perspectives from the americas. *Transportation*, 42(3):519–536, 2015.

[83] Keiko Shimazaki, Masahiro Kuwahara, Akira Yoshioka, Yukiko Homma, Masaki Yamada, and Akira Matsui. Development of a simulator for one-way EV sharing service. In *20th ITS World Congress ITS Japan*, 2013.

[84] Heinz Spiess and Michael Florian. Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological*, 23(2):83–102, 1989.

[85] Adela Spulber, Eric P. Dennis, Richard Wallace, and Michael Schultz. The impact of new mobility services on the automotive industry. *Center for Automotive Research*, pages 1–56, 2016.

[86] Waiyuen Szeto and Yu Jiang. Transit route and frequency design: Bi-level modeling and hybrid artificial bee colony algorithm approach. *Transportation Research Part B: Methodological*, 67:235–263, 2014.

[87] Yuichi Takano, Keisuke Nanjo, Noriyoshi Sukegawa, and Shinji Mizuno. Cutting plane algorithms for mean-cvar portfolio optimization with nonconvex transaction costs. *Computational Management Science*, 12(2):319–340, 2015.

[88] Yingzi Tan and Rui Deng. China's first autonomous bus line debuts. `http://www.chinadaily.com.cn/a/202104/16/WS6078e3efa31024ad0bab5ed5.html`, Apr 2021. [Online; accessed 1-September-2021].

[89] Qingyun Tian, Yunhui Lin, and David Z.W. Wang. Autonomous and conventional bus fleet optimization for fixed-route operations considering demand uncertainty. *Transportation*, pages 1–29, 2020.

[90] Dean H. Uyeno and Keith A. Willoughby. Transit centre location-allocation decisions. *Transportation Research Part A: Policy and Practice*, 29(4):263–272, 1995.

[91] Quentin K. Wan and Hong K. Lo. A mixed integer formulation for multiple-route transit network design. *Journal of Mathematical Modelling and Algorithms*, 2(4):299–308, 2003.

[92] Wikipedia contributors. List of carsharing organizations — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=List_of_carsharing_organizations&oldid=1033728191`, 2021. [Online; accessed 5-September-2021].

[93] Keith A. Willoughby. A mathematical programming analysis of public transit systems. *Omega*, 30(3):137–142, 2002.

[94] Keith A. Willoughby and Dean H. Uyeno. Resolving splits in location/allocation modeling: A heuristic procedure for transit center decisions. *Transportation Research Part E: Logistics and Transportation Review*, 37(1):71–83, 2001.

[95] Masaki Yamada, Masashi Kimura, Naoki Takahashi, and Akiko Yoshise. Optimization-based analysis of last-mile one-way mobility sharing. *Department of Policy and Planning Sciences Discussion Paper Series*, 1353, 2018.

[96] Bo Yang, Takayuki Ando, and Kimihiko Nakano. Pilot tests of automated bus aiming for campus transportation service. In *2020 5th International Conference on Universal Village (UV)*, pages 1–5, 2020.

[97] Jin Y. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, 27(4):526–530, 1970.

[98] Eileen Yu. First commercial autonomous bus services hit Singapore roads. `https://www.zdnet.com/article/first-commercial-autonomous-bus-services-hit-singapore-roads/`, Jan 2021. [Online; accessed 1-September-2021].

[99] Rabih Zakaria. *Optimization of the Car Relocation Operations in One-way Carsharing Systems*. PhD thesis, Université de Technologie de Belfort-Montbéliard, 2015.

[100] Shahriar A. Zargari, Hassan Khaksar, and Navid Kalantari. Bus fleet optimization using genetic algorithm: A case study of Mashhad. *International Journal of Civil Engineering*, 11(1):43–52, 2013.

[101] Kai Zhang, Yuichi Takano, Yuzhu Wang, and Akiko Yoshise. Optimizing the strategic decisions for one-way station-based carsharing systems: A mean-cvar approach. *IEEE Access*, 9:79816–79828, 2021.

[102] Meng Zhao, Xiaopeng Li, Jiateng Yin, Jianxun Cui, Lixing Yang, and Shi An. An integrated framework for electric vehicle rebalancing and staff relocation in one-way carsharing systems: Model formulation and lagrangian relaxation-based solution approach. *Transportation Research Part B: Methodological*, 117:542–572, 2018.

Appendix A

Optimal strategies assignment model

Passenger assignment or route assignment concerns the selection of routes between origins and destinations in transportation networks. The demand can be assigned to the corresponding shortest paths for the car mode, called the 'all-or-nothing' assignment. However, this is not a proper solution in the transit network due to waiting for the buses (waiting cost). To capture this situation, Spiess and Florian [84] proposed an optimal strategies assignment model for transit networks.

Consider a transit network $G = (N, A)$, where $n \in N$ are nodes (real or dummy bus stops) and $a \in A$ are links between two nodes. Let A_n^+ (A_n^-) denote the set of outgoing (incoming) arcs at $n \in N$. Figure A.1 gives an example of a transit network (or called trajectory network).For the transit route choice problem, a strategy to reach destination node d is defined by a partial network $G_d = (N, \bar{A})$. For node $n \in N$, $\bar{A}_n^+ = A_n^+ \cap \bar{A}$. For example, if $d =$ Y, in the partial network G_Y, $N =${X,X2,Y2,Y} and $\bar{A}=$ {(X,X2),(X2,Y2),(Y2,Y)} where (X,X2) is a waiting arc, (X2,Y2) is an in-vehicle arc, and (Y2,Y) is a destination arc.

Given a strategy $\bar{A}$, the demand g_n at node $n \in N$ is assigned to the network, which yields arc volumes $v_a, a \in A$. The volume V_n at a node $n \in N$ can be written into the sum of incoming volume and the demand generating at that node.

$$V_n = \sum_{n \in A_n^-} v_a + g_n \tag{A.1}$$

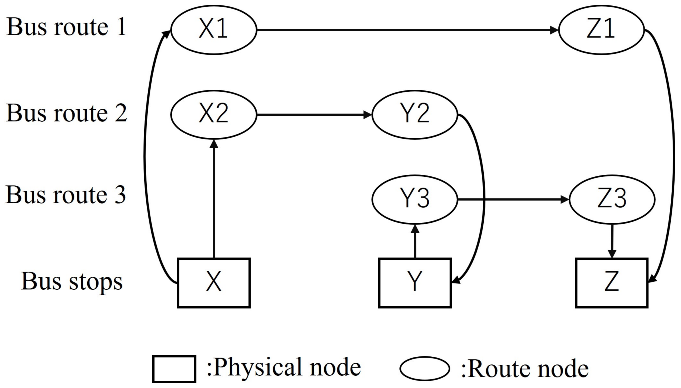


Figure A.1: Example of a transit network

The node volume V_n is assigned to the outgoing arcs based on their link probability.

$$v_a = P_a(\bar{A}_n^+)V_n, a \in A_n^+, \forall n \in N \tag{A.2}$$

where $P_a(\bar{A}_n^+)$ is the probability that arc a is served first among the set $\bar{A}_n^+$.

If the interarrival times of the buses follow exponential distribution, then

$$P_a(\bar{A}_n^+) = \frac{f_a}{\sum_{a' \in \bar{A}^+} f_{a'}}, \forall a \in \bar{A}^+ \tag{A.3}$$

$$W(\bar{A}_n^+) = \frac{1}{\sum_{a \in \bar{A}^+} f_a} \tag{A.4}$$

where f_a is the mean frequency of link $a \in \bar{A}_n^+$, $W(\bar{A}_n^+)$ is the expected waiting time for the first bus arriving at node n. The proof can be found in [48, 58].

We aim to find the optimal strategy $\bar{A}^\star$ that minimizes the expected total travel time including waiting time, so the assignment model can be given as follows. We use binary variable $x_a, a \in A$ to represent whether arc a is in $\bar{A}^\star$.

$$\min \sum_{a \in A} c_a v_a + \sum_{n \in N} \frac{V_n}{\sum_{a \in A_n^+} f_a x_a} \tag{A.5a}$$

subject to:

$$v_a = \frac{f_a x_a}{\sum_{a' \in A_n^+} f_{a'} x_{a'}} V_n, \qquad \forall a \in A_n^+, n \in N \tag{A.5b}$$

$$V_n = \sum_{a \in A_n^-} v_a + g_n, \qquad \forall n \in N \tag{A.5c}$$

$$V_n \in \mathbb{R}_+, \qquad \forall n \in N \tag{A.5d}$$

$$v_a \in \mathbb{R}_+, \qquad \forall a \in A \tag{A.5e}$$

$$x_a \in \{0, 1\}, \qquad \forall a \in A \tag{A.5f}$$

The objective is to minimize total travel time on arcs (the first term) and total waiting time at nodes(the second term). By summing constraints (A.5b) for all $a \in A_n^+$ and considering constraints (A.5c), we can obtain the flow conservation, that is, constraints (A.7b).

Let

$$w_n = \frac{V_n}{\sum_{a \in A_n^+} f_a x_a}, \tag{A.6}$$

then, we can reformulate the problem (A.5) and obtain the equivalent problem (A.7), see [84].

$$\min \sum_{a \in A} c_a v_a + \sum_{n \in N} w_n \tag{A.7a}$$

subject to:

$$\sum_{a \in A_n^+} v_a - \sum_{a \in A_n^-} v_a = g_n, \qquad \forall n \in N \tag{A.7b}$$

$$v_a \leq f_a w_n, \qquad \forall a \in A_n^+, n \in N \tag{A.7c}$$

$$v_a \in \mathbb{R}_+, \qquad \forall a \in A \tag{A.7d}$$

Appendix B

General mean-CVaR model

Conditional value-at-risk (CVaR) was first proposed by Rockafellar and Uryasev [76] as a downside risk measure to quantify tail losses. In comparison with the traditional risk measure value-at-risk (VaR), CVaR has more attractive mathematical properties (e.g., subadditivity, convexity; see [74]). More importantly, CVaR can illustrate the losses exceeding VaR, but VaR does not control such losses.

Let $\mathscr{L}(\boldsymbol{x}, \boldsymbol{y})$ denote the loss function with respect to a decision vector $\boldsymbol{x} \in X \subset \mathbb{R}^n$ and uncertain vector $\boldsymbol{y} \in \mathbb{R}^m$ and $p(\boldsymbol{y})$ denotes the probability density function associated with $\boldsymbol{y}$. For a fixed decision vector $\boldsymbol{x}$, the cumulative distribution function of the loss can be written as:

$$\psi(\boldsymbol{x}, \alpha) = \int_{\mathscr{L}(\boldsymbol{x}, \boldsymbol{y}) \leq \alpha} p(\boldsymbol{y})\ d\boldsymbol{y}. \tag{B.1}$$

Given a confidence level β, the β-VaR associated with the decision vector $\boldsymbol{x}$ is as follows:

$$\mathrm{VaR}_\beta(\boldsymbol{x}) = \min\{\alpha \in \mathbb{R} : \psi(\boldsymbol{x}, \alpha) \geq \beta\}. \tag{B.2}$$

Three values of β are commonly considered: 90%, 95%, and 99%. The β-CVaR associated with the decision vector $\boldsymbol{x}$ is defined as:

$$\mathrm{CVaR}_\beta(\boldsymbol{x}) = \frac{1}{1-\beta} \int_{\mathscr{L}(\boldsymbol{x}, \boldsymbol{y}) \geq \mathrm{VaR}_\beta(\boldsymbol{x})} \mathscr{L}(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{y}) d\boldsymbol{y}, \tag{B.3}$$

which is the conditional expectation of the loss that is beyond VaR_β . It is difficult to deal with (B.3) directly, so Rockafellar and Uryasev [76] developed a simpler auxiliary function:

$$\mathscr{F}_\beta(\boldsymbol{x}, \alpha) = \alpha + \frac{1}{1-\beta} \int_{\boldsymbol{y} \in \mathbb{R}^m} [\mathscr{L}(\boldsymbol{x}, \boldsymbol{y}) - \alpha]_+ p(\boldsymbol{y}) d\boldsymbol{y} \tag{B.4}$$

where $[x]_+ = \max\{x, 0\}$ for $x \in \mathbb{R}$, and proved

$$\text{CVaR}_\beta(\boldsymbol{x}) = \min_{\alpha \in \mathbb{R}} \mathscr{F}_\beta(\boldsymbol{x}, \alpha). \tag{B.5}$$

For practical applications, we often consider the following scenario-based approximation, i.e., (B.6), with a number of scenarios in the name of $\boldsymbol{y}^s$ for s in the scenario set S to avoid numerical difficulties caused by the integration in (B.4).

$$\mathscr{F}_\beta(\boldsymbol{x}, \alpha) \approx \alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} \left[\mathscr{L}\left(\boldsymbol{x}, \boldsymbol{y}^s\right) - \alpha\right]_+ \tag{B.6}$$

The system operators usually try to minimize the risk while maximizing the return (or minimizing the loss). In this case, we can consider the mean-CVaR model, taking both return and risk into account. Krokhmal et al. [55] illustrated three equivalent formulations of the mean-CVaR model. A typical one among them is

$$\min_{\boldsymbol{x}, \alpha} \left\{\lambda \mathbb{E}[\mathscr{L}(\boldsymbol{x}, \boldsymbol{y})] + (1-\lambda) \mathscr{F}_\beta(\boldsymbol{x}, \alpha)\right\}, \tag{B.7}$$

where λ is the weight value, ranging from 0 to 1, and $\mathbb{E}[\cdot]$ is the expectation function. Considering discrete scenarios with the same probability, we can reformulate problem (B.7) in an approximate form:

$$\min_{\boldsymbol{x}, \alpha} \left\{ \frac{\lambda}{|S|} \sum_{s \in S} \mathscr{L}\left(\boldsymbol{x}, \boldsymbol{y}^s\right) + (1-\lambda) \left(\alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} \left[\mathscr{L}\left(\boldsymbol{x}, \boldsymbol{y}^s\right) - \alpha\right]_+ \right) \right\}. \tag{B.8}$$