UNIVERSITY OF LEEDS

This is a repository copy of *A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/175285/

Version: Accepted Version

Article:

Fu, C, Zhu, N, Ma, S et al. (1 more author) (2022) A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems. European Journal of Operational Research, 298 (3). pp. 915-938. ISSN 0377-2217

https://doi.org/10.1016/j.ejor.2021.06.014

© 2021, Elsevier. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can’t change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose
university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Fu, C., Zhu, N., Ma, S., and Liu, R. (2021) A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems. European Journal of Operational Research. Available online at: https://doi.org/10.1016/j.ejor.2021.06.014.

A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems

Chenyi Fu[a], Ning Zhu[a,*], Shoufeng Ma[a], Ronghui Liu[b]

[a]*Institute of Systems Engineering, College of Management and Economics, Tianjin University, Tianjin 300072, China*
[b]*Institute for Transport Studies, University of Leeds, UK*

Abstract

A bike-sharing system is a shared mobility mechanism that provides an alternative transportation mode for short trips with almost no added travel speed loss. However, this model's low usage ratio and high depreciation rate pose a risk to the sustainable development of the bike-sharing industry. Our study proposes a new integrated station location and rebalancing vehicle service design model. This model aims to maximize daily revenue under a given total investment for station locations and bike acquisition. To address demand ambiguity due to possible bias and loss of data, we present a two-stage robust optimization model with a demand-related uncertainty set. The first stage of our model determines the station locations, initial bike inventory, and service areas of rebalancing vehicles. In contrast to the literature, which either simplifies the rebalancing process as an inventory transshipment problem or formulates it as a complex dynamic bike rebalancing problem, we assign each rebalancing vehicle to a service area composed of several specified stations. An approximate maximum travel distance for each rebalancing vehicle is also designed and constrained to ensure that the rebalancing operation can be performed within each period. In the second stage, our model optimizes the daily fleet operation and maximizes the total revenue minus the rebalancing cost. To solve our model, we design a customized row generation approach. Our numerical studies demonstrate that our algorithm can efficiently obtain exact solutions in small instances. For a real-size problem, the nearly optimal solutions of our model also reveal a high-quality worst-case performance with a small loss in mean performance, particularly when the value of the budget ratio (that is, the average number of bikes per station) is at a medium level. Moreover, the distribution of service areas depends on the bike supply and demand level at each station. The optimal fleet rebalancing operation does not have to be confined to one geographical area. Furthermore, our robust model can achieve larger mean and worst-case revenues and a higher revenue stability than a stochastic model with a small data set.

Keywords: Transportation, Bike-sharing system, Station location problem, Robust optimization, Row generation

1. Introduction

The urbanization process drives rapid economic development as well as serious traffic congestion, noise, and air pollution (Dunlap and Jorgenson, 2012). To maintain the sustainable development of metropolises, the emerging sharing economy promotes innovations in shared mobility systems, such as peer-to-peer sharing (e.g., Turo), ride-sharing (e.g., Uber and Didi) and bike/car sharing (e.g., Mobike and Car2go). Unlike other shared mobility systems, bike-sharing systems have already undergone several generational transformations, including free-bike, coin-deposit, and information-technology-based systems, over the past half century (Demaio, 2009). Recently, station-based bike-sharing systems, which allow users to pick up and drop off bikes at stations, have gradually been evolving. According to the Bike Sharing World Map (www.bikesharingworld.com), as of March 2019, there were approximately 1,950 operational bike-sharing programs and approximately 14,860,200 bikes in service worldwide. For example, one of the most popular and large-scale public bike-sharing programs,

[*]Corresponding author

Email addresses: chenyi_fu_201509@163.com (Chenyi Fu), zhuning@tju.edu.cn (Ning Zhu), sfma@tju.edu.cn (Shoufeng Ma), R.Liu@its.leeds.ac.uk (Ronghui Liu)

namely, Velib in Paris, France, has approximately 14,500 bikes and 1,230 stations. Some studies note that bike-sharing systems can serve a similar purpose as private cars for short trips, such as shopping and commuting, without added travel speed loss in downtown settings (Jensen et al., 2010). On April 12, 2017, Mobike, one of the largest bike-sharing firms in China, issued a white paper (www.sohu.com/a/133766880_585110) suggesting that on 92.9 percent of days in Beijing, using public transportation and bikes is faster than using only private cars, particularly when the travel distance is less than 5 km during peak hours. Similar results have been reported for Shanghai. In addition, cycling is considered an environmentally friendly mode of transport. The white paper indicated that use of the bike-riding mode has reduced carbon emissions by 540 thousand tons in addition to reducing PM 2.5 by 4.5 billion microgrammes, which is equivalent to saving approximately 460 million litres of gasoline, since Mobike began operations.

However, a low utilization rate and the substantial waste of public space caused by the inconvenience of searching for available docks and additional walking time to destinations restrict the development of bike-sharing systems. Shu et al. (2013) state that higher investment in station construction and lower bike usage hinder the bike-sharing systems in various cities of China compared with systems in Europe and Canada. This problem has severely affected bike-sharing systems in China because most bike-sharing firms in China are not nonprofit organizations and are at risk of financial failure. For example, in 2018, the IPO documents of Meituan revealed that Mobike posted a total loss of 407 million RMB from April 4 to April 30, 2019, which was mainly composed of a total revenue of 107 million RMB, depreciation costs of 396 million RMB, and daily operation costs of 158 million RMB. Ofo, one of the largest bike-sharing firms in China, filed for bankruptcy protection in 2019. It is important for these firms to make reasonable strategic and operational decisions to maximize their revenue. In addition, station locations play a critical role in the performance of bike-sharing systems. Bike stations do not occupy a large space, and there are a fairly large number of candidate locations to choose from. Hence, we can optimize bike station locations without restrictions due to land authorizations and other policy-related factors.

Recent studies show that the choice of bike station locations plays a fundamental role in the performance of bike-sharing systems (Wang and Lindsey, 2019; Kabra et al., 2019). Two of the most important factors that affect system performance are accessibility and availability, in addition to exogenous factors such as temperature, precipitation, and wind speed (Kabra et al., 2019). Accessibility is determined by how far a user must walk to reach a station. Availability is defined as the likelihood of finding a bicycle at a station. The findings of these authors show that almost 80% of bike usage comes from stations within 300 m from the origin (Kabra et al., 2019). Therefore, station location is the primary critical factor when designing bike-sharing systems. In addition, both accessibility and availability are closely related to two fundamental decisions, i.e., choosing station locations and setting inventory. Compared with that of other public transportation systems, such as buses and subways, the capacity of each station in a docking station system ranges from 10 to 50 docks, which does not warrant strict administrative authorization requirements. Recently, with the help of GPS and Bluetooth technologies, stations have become virtual areas without dock investment in free-float bike-sharing systems that are not large on the ground and are not greatly restricted by land use.

There are several reasons why we should consider the bike station location decision and rebalancing vehicle service design problems simultaneously. After the station location, designing a dynamic rebalancing service is the most commonly used method to update the inventory level of each station during a period. Intuitively, optimizing the bike station location decision and the rebalancing vehicle service design problems simultaneously could be better than considering them independently. The first reason is that the station location decision greatly influences the efficiency of the vehicle rebalancing operation. If the station locations are chosen to be overly dispersed and the demand distribution is overlooked, this may increase the rebalancing travel distance and cause a high supply-demand imbalance in each station, significantly increasing the cost of the rebalancing operation. In other words, more rebalancing vehicles may be needed. Second, the recent development of bike sharing technology has enabled mobile bike docks. This means that bike station docks are sometimes able to move from one location to another. This new feature makes it possible to achieve a higher system performance. It also indicates the importance of the station location problem. Third, in the context of free-floating bike-sharing systems, a station location is a virtual area without real docks where the firm wants the user to return bikes. This makes it easier for the firm to reselect bike station locations such that a better rebalancing cost and greater total system revenue can be obtained. Fourth, in the era of big data, bike system usage data may be available, making it possible to simultaneously consider station locations and the closely related vehicle rebalancing service design problem.

In our paper, we jointly consider strategic, tactical and operational decisions for bike-sharing systems. We assume that there are several zones in a geographical area and that each zone has a candidate station (see the upper-left figure in Fig. 1). At the strategic and tactical levels, our model determines the location of bike stations (see the middle-left figure in Fig. 1) and the number of bikes allocated to each station under a given initial investment. At the operational level, we aim to determine the covered demand and the number of rebalanced bikes to maximize the daily operational net revenue. Studies in the literature either formulate the rebalancing process as a vehicle routing problem with delivery and pick-up (VRPDP) or simplify it as a transshipment problem. In the upper-right and middle-right figures in Fig. 1, we show instances of the VRPDP and transshipment problem, respectively, where a positive (negative) scalar near the station implies that this station is oversupplied (undersupplied). In this example, the VRPDP determines three routes to rebalance the inventories (i.e., 7-4-2-5-7, 8-3-6-11-8 and 10-12-13-10). The scalars on the routes are the numbers of bikes in each vehicle. For the transshipment problem, we only determine the number of bikes rebalanced between two stations. Note that owing to the distribution of stations and the various bike inventory levels, zones in a service area may not be adjacent (see the service area highlighted in orange).

Regarding the modelling of the rebalancing process, both the VRPDP and transshipment approaches have their own advantages and disadvantages. The results of the VRP model provide an exact routing plan for each rebalancing vehicle. However, if the resultant route is computed from a deterministic model, it lacks flexibility to uncertainty in the potential model input. In addition, a recent study has shown that drivers have room to adjust the given routes based on their own experience and traffic conditions (Liu et al., 2018). From a computational perspective, it is also very challenging to generate routes. The transshipment problem modelling technique oversimplifies the rebalancing process and can only obtain the results of the quantity of bikes moved between each pair of stations. It does not consider the limited number of rebalancing vehicles or the limited travel time of a vehicle in each period. We stand between these two popular methods and provide a computationally feasible approach.

Our work presents a new model and assigns each rebalancing vehicle to a service area; this area contains several stations (see the lower-right figure in Fig. 1). The travel cost for each service area is approximated and is assumed to be less than or equal to a given threshold. Thus, a vehicle in each service area is able to complete the rebalancing operation in each period. The system is modelled as a spatiotemporal network. This problem is also a stochastic dynamic program considering demand uncertainty. To address the issues of limited data and the computational intractability dilemma due to the curse of dimensionality, we employ a two-stage robust optimization framework to obtain an approximation solution considering spatial demand correlations. We use this framework to approximate the multistage dynamic optimization results (Weikl and Bogenberger, 2013; Lu et al., 2017). More details of our problem description and formulation are given in section 3. The main contributions of our work are threefold.

First, the rebalancing process embedded in the station location problem can be considered a compromise between two rebalancing formulations, that is, a routing-based and a transshipment-based program. Our model assigns stations to service areas. Each service area is assigned to only one rebalancing vehicle. We impose a restriction whereby bikes can only be rebalanced among stations in the same service area. To ensure that all stations in each service area are served within a prespecified time period, we use a minimum spanning tree to approximate the maximum travel distance, which is an upper bound of the routing distance. This maximum travel distance should not exceed a given threshold, which ensures that all stations can be visited in each period. Based on the above idea, we propose two models: a deterministic model and its robust counterpart.

Second, due to the inaccuracy of demand and the computational intractability of a stochastic dynamic model, we employ a two-stage robust optimization (max-min-max) framework that considers the spatial correlations among demands to approximate the multistage dynamic solutions. The robust approach is capable of modelling in the absence of exact demand information (e.g., the exact probability density function of demand). The approach also provides a solution with high mean and worst-case revenue compared to that of the stochastic programming approach. The first stage determines the positions of stations and service areas, while the second stage ensures the flow conservation of the served demand and rebalanced bikes. The inner program of our model is transformed into a minimum disjoint bilinear program. To solve this model, we design a customized row generation approach. We use duality theory and the Kuhn-Tucker condition so that the inner program can be reformulated as a linear min-max program and further as a mixed-integer linear model. Similar to Benders decomposition, this mixed-integer linear program (MILP) of the inner problem is treated as a subproblem, while the rest of the objective functions and constraints in the primal model form the master problem. Then, we solve

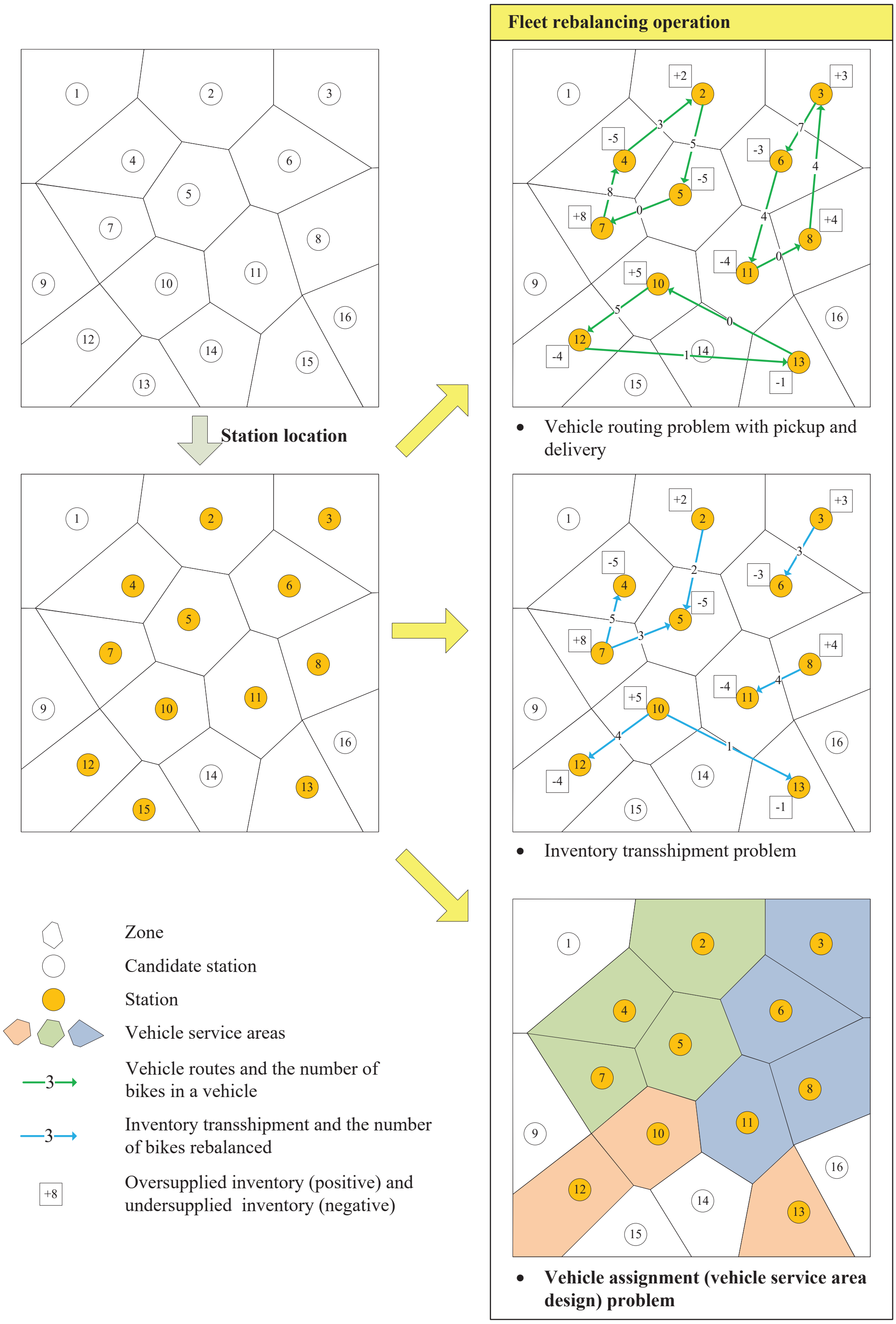

Fleet rebalancing operation
Station location
Vehicle routing problem with pickup and delivery
Inventory transshipment problem
Vehicle assignment (vehicle service area design) problem
Zone
Candidate station
Station
Vehicle service areas
Vehicle routes and the number of bikes in a vehicle
Inventory transshipment and the number of bikes rebalanced
Oversupplied inventory (positive) and undersupplied inventory (negative)
+8

Figure 1: Station locations and three planning aspects of the rebalancing operation

the master problem and subproblem iteratively, and at each iteration, a valid cut is generated and added to the master problem based on the solution of the subproblem. The finite convergence of this approach is guaranteed.

Third, we use open data and design a series of experiments to analyse the performance of our model and approach. Some instances based on a small-scale network are used to verify the efficiency of the approach. Compared with a scenario-based model, in which efficiency is highly dependent on scenario combinations and which cannot guarantee that we will obtain the optimal solution, our approach can reach the global optimum within a short computational time. In addition, experiments based on a real-size network demonstrate that the robust solutions can improve the worst-case performance by approximately 10% with a small price of mean performance loss at a mid-level budget ratio (the average number of bikes at each station). The solutions of the robust and deterministic models offer different bike station distributions. More zones have stations in the robust model than in the deterministic model over all instances to ensure greater demand coverage.

The remainder of this paper is organized as follows: Section 2 reviews relevant studies, including problems in bike-sharing systems and optimization approaches for uncertainty. All details of our formulation and the associated algorithm are described in sections 3 and 4, respectively. In section 5, we employ open real data to test our model and algorithm. Finally, we conclude our work in section 6.

2. Literature review

There is a rapidly growing body of literature on bike and car-sharing systems. We mainly focus on studies of bike-sharing systems at the strategic and operational levels as well as studies of car-sharing systems. For details on the history of bike-sharing industries, we refer the reader to Demaio (2009) and Shaheen et al. (2010). Laporte et al. (2015) recently summarized bike-sharing problems at the strategic, tactical, and operational levels.

2.1. Station location problem

One major stream of literature focuses on strategic bike station location design. The decision variables in these problems are concerned with locations and station capacity. Lin and Yang (2011) study strategic planning for bike-sharing systems and jointly determine the optimal stations, bike lane locations, and bike stocks. Garcìa-Palomares et al. (2012) and Park and Sohn (2017) both employ the p-median and maximum coverage model to determine the bike stations. To investigate the potential trip demand, these two papers use GIS-based methodology and real taxi trajectory data. Çelebi et al. (2018) use a queueing model to measure the service levels and then employ a set-covering model to locate stations.

Some studies further integrate the tactical and operational decisions (i.e., the inventory problem and the fleet rebalancing operation) into the bike station location problem. Lin et al. (2013) investigate a station location model with bike inventory costs and design a greedy heuristic method to obtain a near-optimal solution. Frade and Ribeiro (2015) formulate a MILP to maximize the covered demand considering the balance constraint of annual costs and revenue. This model determines station locations, initial station inventories, and bike rebalancing. In contrast to the maximum system utility and minimum total budget used in the above model-s, Nair and Miller-Hooks (2014) demonstrate that users will not follow the system-optimal assignment policy since users are likely to access systems only if their travel utilities can be improved. The authors present an equilibrium station location model formulated as a bi-level program. The objective function of the upper-level decision is to maximize the revenue of demand trips, while in the lower-level problem, users react to a system configuration by minimizing the sum of their travel time and waiting time. Maggioni et al. (2019) adopt two-stage and multi-stage stochastic programming approach to determine station location and rebalancing decisions. They also consider that users may return bikes to other stations since there may be no available spaces at the destination station.

Similar to bike-sharing systems, the literature also investigates station locations and the service region design of car-sharing systems integrated with tactical and operational decisions. Boyacıet al. (2015) present a MILP model incorporating a one-way car-sharing station location model considering capacity and fleet size subject to rebalancing and recharging requirements. The authors apply the branch-and-bound algorithm to obtain the optimal solution. Kaspi et al. (2014) use a MILP model to design parking reservation policies in one-way car-sharing systems with the associated user behaviour responses. Brandstätter et al. (2017) optimize the station locations of car-sharing systems using a time-dependent two-stage stochastic optimization approach. He et al. (2017) maximize the profit and optimize the service regions of free-float electric car-sharing systems with uncertain customer adoption behaviour in response to the service coverage. The queueing network method is used to describe the rebalancing and recharging operations. Lu et al. (2017) consider both station-based and free-float systems and present a two-stage stochastic integer program to model uncertain demand. In the first stage of the model, the fleet and parking lots are investment decisions, while fleet operations are modelled to maximize revenue based on the network flow technique in the second stage. Chang et al. (2017) design an integer linear program for one-way and round-trip car-sharing station locations that maximizes the total profit and minimizes the cost of rebalancing and maintenance under a limited investment cost and a given CO_2 emission limit.

2.2. *Bike rebalancing problem*

In addition to the bike station location problem, another major concern for bike-sharing systems is bike rebalancing to effectively meet demand. A common modelling method is to simplify the rebalancing problem as an inventory transshipment problem, which ignores the detailed rebalancing process in the optimization model. Nair and Miller-Hooks (2011) develop a joint-chance-constraint-based stochastic MILP to optimize vehicle rebalancing schemes to ensure a high satisfaction rate of random demand. Shu et al. (2013) analyse the demand distributions of short subway trips in Singapore and apply a stochastic network flow model to study the bike rebalancing problem. He et al. (2020) construct a multistage distributionally robust program for vehicle rebalancing to approximate the solution to avoid loss of information and the curse of dimensionality of traditional stochastic dynamic models. A similar idea is adopted in strategic station location problems so that the rebalancing process is usually formulated in terms of flow conservation constraints. However, this simplified formulation is unrealistic and does not consider the capacity restrictions of rebalancing vehicles or the accessibility requirements between two stations within a unit period.

Other notable studies view the bike rebalancing process as a VRPDP that can be further categorized into static and dynamic rebalancing problems. The former type of problem assumes that repositioning is carried out only at night (see Tal and Ofer, 2013; Forma et al., 2015; Li et al., 2016; Szeto et al., 2016; Ho and Szeto, 2017; Bulhões et al., 2018). Erdoğan et al. (2014) relax the assumption that optimal inventory levels are located in intervals rather than being single values and use Benders decomposition to solve small-scale instances. Li et al. (2016) investigate the static rebalancing problem with heterogeneous docks and bikes in a single system and use a genetic algorithm to obtain solutions. Bulhões et al. (2018) consider the restriction capacities and service times of the rebalancing process without limiting the visit times for each station. Pal and Zhang (2017) first minimize the total rebalancing cost for free-floating systems to ensure that all dispersed bikes are collected at specified stations. Lei and Ouyang (2018) use the continuous approximation approach for demand rebalancing to solve large-scale problems.

In addition to the static rebalancing strategy, the dynamic rebalancing problem allows bikes to be redistributed in the daytime to satisfy real-time demand (see Contardo et al., 2012; Kloimüllner et al., 2014; Pfrommer et al., 2014; Schuijbroek et al., 2017; Shui and Szeto, 2017; Zhang et al., 2017; Legros, 2019). Contardo et al. (2012) incorporate column generation and Benders decomposition into a new approach to solve mid-size problems. Pfrommer et al. (2014) and Haider et al. (2018) suggest real-time price incentives to induce demand and reduce the need for rebalancing. Shui and Szeto (2017) optimize unserved demand and carbon emissions from rebalancing vehicles.

2.3. *Uncertainty and robust optimization*

Since firms cannot obtain trip information before demand is covered, demand uncertainty caused by changeable weather, accidents, geographical information, and travel preferences is a major concern for strategic station location decisions. The current literature on the station location problem for bike-sharing systems has paid less attention to potential uncertainty in demand than to other sources of uncertainty. To address demand uncertainty, Maggioni et al. (2019) use two-stage and multistage stochastic programming to design the station locations and allocate the spillover of demand at a station to nearby stations. Similarly, Dell'Amico et al. (2018) and Warrington and Ruchti (2019) employ stochastic programming approaches in dynamic bike rebalancing problems.

When the probability distribution of uncertainty is unknown and ambiguous, a robust optimization technique is more suitable than stochastic programming. We refer readers to Ben-Tal et al. (2009) and Bertsimas et al. (2011) for a review of various robust optimization approaches. Several novel approaches considering only partial distribution information, namely, distributionally robust optimizations, are further studied (Delage and Ye, 2010; Goh and Sim, 2010; Wiesemann et al., 2014; Bertsimas et al., 2018). Our formulation follows a robust optimization framework where the demand falls within a given polyhedron. Substantial numbers of papers based on polyhedral uncertainty sets are discussed both in terms of methodology (Ben-Tal and Nemirovski, 1997; Ben-Tal and Nemirovski, 1998; Bertsimas and Sim, 2004) and applications (Liu and Song, 2017; Chan et al., 2018).

Methodologically, our work is similar to that of Chan et al. (2018); both consider a two-stage robust optimization model. To solve the program, both studies design customized row generation algorithms. The motivation in designing the algorithm in their work is to address the large-scale problem efficiently. However, in our study, complex constraints make our model challenging, and it cannot be reformulated into a deterministic equivalent MILP. Thus, we present a row generation approach to solve the primal problem. Further discussion of row and column generation for robust optimization can be found in Zeng and Zhao (2013).

2.4. *Location routing problem*

Since our model jointly considers the station locations and the rebalancing operation, it is closest to a variant of the location-routing problem (LRP), the capacitated LRP. There are many variants of the standard LRP, such as the dynamic LRP with multiple planning periods, where information becomes available over time (Albareda-Sambola et al., 2012); the periodic LRP, which determines the periods of visiting the customers (Francis et al., 2008); the stochastic LRP, which considers uncertain travel time and demand (Albareda-Sambola et al., 2007); and the LRP with pickup and delivery (Karaoglan et al., 2012). Similar to most vehicle routing problems, large-scale instances of the LRP cannot be solved by state-of-the-art commercial solvers. We refer the reader to Drexl and Schneider (2015) for more on the variants and extensions of the location-routing problem.

2.5. *Comparative analysis*

In the context of bike-sharing systems, most of the current studies consider either the station locations or the rebalancing process planning. Regarding the station location literature, the most considered factors are the covered demand, station capacity, bike inventory, and total budget of the system. These factors appear in either the objective functions or constraints. In detail, most of the extant studies formulate the station location problem as a classic facility location model. On the other hand, daily operational decisions are usually considered to be based on a flow conservation framework. We differ from the literature in that we simultaneously optimize bike station locations and the distribution of the service areas of the rebalancing vehicles. In our study, a service area with limited range is used to ensure that the daily dynamic rebalancing operation of each vehicle can be finished in a single period. When the range of the service area is assumed to be large enough, rebalancing operations can be performed between any two stations; this turns out to be a common assumption in the existing literature. Hence, our model is a more generalized bike station location model than previous models in the literature.

Two modelling techniques are often employed to model the rebalancing process: vehicle routing problems with pickup and delivery and transshipment problems. These two methods usually consider the capacity, the number of depots, multiple periods, and so on. Our study focuses on strategic-level decisions, so a detailed routing plan cannot satisfy dynamic and uncertain demand. Hence, we use the service areas of rebalancing vehicles to ensure that all possible routes in each service area can be served in one period. In addition, demand uncertainty is considered, and a two-stage robust optimization framework is employed. We also design an efficient algorithm to solve a problem of practical size using realistic data.

3. Mathematical formulation

In this section, we first describe our problem and define related notations in section 3.1; then, we present our two-stage robust model in section 3.2 and design the demand uncertainty set in section 3.3.

3.1. *Problem description*

We consider a station location and initial inventory allocation problem for station-based bike-sharing systems under a given total investment. Suppose that the set of demand zones is known based on historic data and is denoted by $\mathcal{N}$. Each zone is assumed to have a candidate station location to cover the demand in that zone. We use the binary variable x_i to represent whether there is a station in zone i; $x_i = 1$ implies that there is a station in zone i, and $x_i = 0$ otherwise. In our model, we assume there is only one candidate station in each zone. For the case with multiple candidate stations, we can further partition this zone into smaller ones so that each new smaller zone has one candidate station. Hence, our assumption can still be applied. We assume that each station has the same maximum capacity C. This assumption is practical and reasonable because bike-sharing firms employ a GPS-based technique and locate bikes without locks. Hence, the setup costs of the station areas are low, and in this case, the capacity is not a significant decision variable. This assumption can be relaxed by determining the capacity of each station in our model and does not change the form of the model. In practice, a firm conducts fleet operations over a set of periods $\mathcal{T}$. The fleet of vehicles is used to rebalance the bikes between stations.

As mentioned above, the previous literature on the bike station location problem considers only the number of bikes rebalanced between stations, which is embedded into the flow conservation constraints for easy formulation. This modelling approach, namely, the inventory transshipment model, is simple and does not consider a limited number of vehicles or the maximal travel distance per period. In our study, we design a static vehicle assignment problem to ensure that all possible routes for the dynamic rebalancing operation can be served in each period. That is, we assign each vehicle to a *service area* composed of several stations. The figure in the lower right of Fig. 1 provides an example assigning eleven stations to three service areas. Based on this assumption, we provide flexibility for drivers in adjusting their rebalancing routes.

To make the above assumption mathematically implementable, we need to formally define the relationship between the stations and service areas. In addition, the maximum travel distance within a service area should also be restricted. A notational system and constraints should be designed to reflect these logical requirements.

We denote the set of rebalancing vehicles as K, indexed by k. Since each service area is covered by one vehicle, we also use k to label the service area, and we define the binary decision variable u_k, which is 1 if vehicle k is assigned to the kth service area and 0 otherwise. For service area k, the binary variable z_{ik} is 1 if station i is in service area k and 0 otherwise. The binary variable λ_{ijk} is 1 if stations i and j are both in service area k and 0 otherwise. Thus, some constraints should be established to connect these decision variables (e.g., u_k, z_{ik} and λ_{ijk}), which are shown in the next section.

Let l_{ij} be the travel distance between candidate stations i and j. To ensure that the maximum travel distance in service area k is not larger than a given threshold $\bar{l}_k$, we assume that each service area has exactly one *centre* and let the binary variable y_{ik} equal 1 if station i is the centre of service area k. For each service area, there is only one centre, which is a special station. Furthermore, the variable l_{ijk} is equal to l_{ij} if stations i and j are both in service area k and station i is its centre, and it is 0 otherwise. The sum of the distance between the centre and each station j in service area k can be easily calculated to approximate the travel distance of each vehicle. This method essentially calculates the distance of a minimum spanning tree constructed by the stations in a service area, which is a 2-approximation of the vehicle routing problem and can be seen as an upper-bound distance of all possible routes. As an example, the red spots in Fig. 2 represent the centres, and the red lines construct three minimum spanning trees and are used to calculate the total distances between the centres and stations.

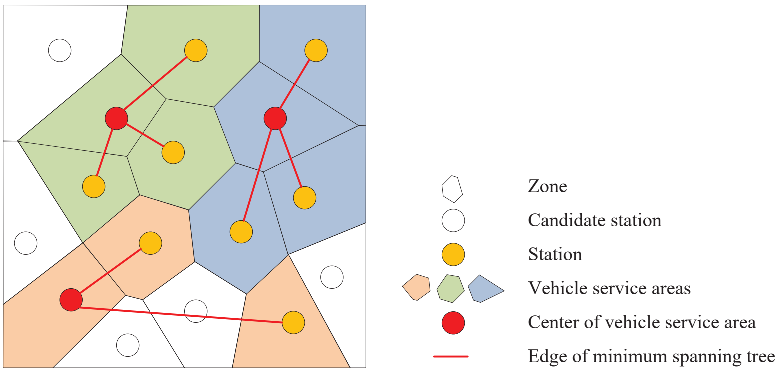


Figure 2: A vehicle assignment scheme (vehicle service area design) for the dynamic rebalancing operation

Considering demand uncertainty, our integrated station location and rebalancing vehicle assignment problem is modelled using a two-stage robust optimization framework. In the first stage, the station locations and initial bike allocation are determined under a given number of station locations B_1 and a bike allocation B_2, as well as the service area design. The fixed operation cost per vehicle is c_2. In the second stage, the objective function is to maximize the revenue minus the rebalancing cost. The second-stage decision variables are mainly concerned with the dynamic inventory levels, satisfied demand decision and fleet rebalancing decision in each period.

We assume that users in zone i can access the bike-sharing system if there is a station with available bikes. Similarly, users ride bikes and return them at destinations that have enough available spaces. For each period t, the total random demand from zones i to j is denoted as $\tilde{d}_{ijt}$. At the beginning of period t, firms are assumed to have information on the current inventory v_{it} at each bike station i and the historical demand. We assume that the rebalancing operations will be finished before determining customer demand. Regarding the rebalancing operations, the number of bikes from station i to j, denoted as the decision variable r_{ijt}, should be determined. The rebalancing cost per bike is c_{1ij}, the value of which depends on the travel distance between stations i and j. The maximum capacity for each rebalancing operation does not exceed Q. The variable w_{ijt} describes the covered demand from zones i to j in period t, which should be no greater than the total demand $\tilde{d}_{ijt}$ and the current bike inventory v_{it} at station i. To simplify our model, if there is no station in zone i, the local demand will be unmet and cannot be covered by bikes in other zones. The reason is that since the service range of each station is usually small (Li et al., 2015; Park and Sohn, 2017; Kabra et al., 2019) and the bike travel distance is generally short, customers may not walk too far from the origin to rent a bike at the second nearest station. The revenue for each trip is f. The objective function of a private bike-sharing firm aims to maximize revenue during the daytime. The notation used in our paper is listed in table 1. Other notation concerning the uncertainty set and solution algorithm will be defined in the following sections. We refer to bold-faced characters, such as $\boldsymbol{x} \in \mathbb{R}^N$, as N-dimensional vectors, and x_i is the ith component of vector $\boldsymbol{x}$.

Table 1: Notation

Parameters of the robust optimization model	
$\mathcal{T}$	set of periods, indexed by t
$\mathcal{N}$	set of candidate stations, indexed by i, j
$\mathcal{K}$	set of vehicles (and service areas), indexed by k
c_{1ij}	rebalancing cost per bike from station i to j
c_2	fixed operation cost of a vehicle
f	revenue per demand trip
B_1	total number of candidate stations
B_2	total number of bike allocations
C	capacity of stations
Q	capacity of rebalancing vehicles
$\bar{l}_{ij}$	distance between stations i and j
$\bar{l}_k$	maximal travel distance of vehicle k
Parameters of the uncertainty set and approaches	
$\mathcal{U}$	uncertainty set of demand
Ω, Ω'	set of demand samples, indexed by ω
$\mathcal{S}$	set consisting of all subsets of OD pairs, indexed by s
$\mathcal{OD}_{st}$	sth subset of OD pairs at period t
$\mathbb{P}$	joint probability distribution of all demand
$\tilde{d}_{ijt}$	uncertain demand from zones i to j at period t
$\overline{d}_{ijt}$	upper bound of demand from zones i to j at period t
$\underline{d}_{ijt}$	lower bound of demand from zones i to j at period t
Δd_{ijt}	interval length of $\tilde{d}_{ijt}$; that is, $\overline{d}_{ijt} - \underline{d}_{ijt}$
Γ_{st}	minimal demand threshold of all OD pairs in $\mathcal{OD}_s$ at period t
κ_t	equals 1 if $t < \lvert\mathcal{T}\rvert$ and 0 otherwise
M	big number
$\boldsymbol{q}$	random weight vector
α	parameter to define the degree of uncertainty
First-stage decision variables of the robust optimization model	
x_i	equals 1 if bike station is set at zone i and 0 otherwise
u_k	equals 1 if vehicle k is used and 0 otherwise
z_{ik}	equals 1 if station i is in service area k and 0 otherwise
y_{ik}	equals 1 if station i is the centre of service area k and 0 otherwise

l_{ijk}	equals the distance between a centre i and station j if they are in service area k and 0 otherwise
λ_{ijk}	equals 1 if stations i and j are both in service area k and 0 otherwise
Y	variable in the MP of the row generation approach
Second-stage decision variables of the robust optimization model	
v_{it}	number of bikes at bike station i in period t
w_{ijt}	covered demand from stations i to j in period t
r_{ijt}	number of bikes rebalanced from station i to station j in period t
Decision variables for row generation approach	
$\boldsymbol{d}, \boldsymbol{\pi}, \boldsymbol{\mu}, \overline{\boldsymbol{\mu}}, \underline{\boldsymbol{\mu}}, \boldsymbol{u}, \overline{\boldsymbol{u}}, \underline{\boldsymbol{u}}$	variables in the subproblem of the row generation approach

3.2. Two-stage robust optimization model

In this section, we propose a two-stage robust model under demand uncertainty. Based on the problem description above, four sets of constraints are defined, namely, investment constraints, station assignment constraints, service area generation constraints, and fleet operation constraints.

The first set of constraints enforces the basic relationship between the station locations and the number of bikes allocated to each station, which is presented below:

$$\sum_{i\in\mathcal{N}} x_i = B_1, \tag{1}$$

$$\sum_{i\in\mathcal{N}} v_{i0} = B_2, \tag{2}$$

$$v_{it} \leq Cx_i, \qquad \forall i \in \mathcal{N},\ t \in \mathcal{T}. \tag{3}$$

Constraints (1) and (2) restrict the total number of station locations and the bike allocation, respectively. Constraint (3) restricts the number of bikes in each period, which is no greater than the station capacity.

The second set of constraints enforces the relationships among the number of rebalancing vehicles, the service areas and the station locations.

$$\sum_{k\in\mathcal{K}} z_{ik} \leq x_i, \qquad \forall i \in \mathcal{N}, \tag{4}$$

$$z_{ik} \leq u_k, \qquad \forall i \in \mathcal{N},\ k \in \mathcal{K}, \tag{5}$$

$$u_k \geq u_{k+1}, \qquad \forall k \in \mathcal{K}, \tag{6}$$

$$\lambda_{ijk} \leq z_{ik}, \qquad \forall i, j \in \mathcal{N},\ k \in \mathcal{K}, \tag{7}$$

$$\lambda_{ijk} \leq z_{jk}, \qquad \forall i, j \in \mathcal{N},\ k \in \mathcal{K}, \tag{8}$$

$$\lambda_{ijk} \geq z_{ik} + z_{jk} - 1, \qquad \forall i, j \in \mathcal{N},\ k \in \mathcal{K}. \tag{9}$$

Constraint (4) ensures that each station is assigned to at most one rebalancing vehicle. Since our model maximizes the revenue of bike-sharing systems and the number of rebalancing vehicles may be limited, we do not require all stations to be assigned to service areas. In practice, we can rebalance the bikes in these stations at night because there is sufficient rebalancing time. Constraint (5) states that station i can be assigned to vehicle k only if vehicle k is used. We do not require that all vehicles be used because of budget limits. As an example, if there are a total of five homogeneous vehicles and only three of them are used, the optimal solutions $\boldsymbol{u} = (1, 1, 1, 0, 0)$ and $\boldsymbol{u} = (1, 1, 0, 1, 0)$ are equivalent. To eliminate a large number of equivalent solutions, constraint (6) ensures that only the first several vehicles will be used. However, when the rebalancing vehicles are heterogeneous, this constraint should be removed. Constraints (7) – (9) are used to show whether stations i and j are both in service area k. That is, only when z_{ik} and z_{jk} are both equal to 1 is λ_{ijk} equal to 1.

The third set of constraints defines the centre of each service area and determines the associated maximum travel distance.

$$y_{ik} \leq z_{ik}, \qquad \forall i \in \mathcal{N},\ k \in \mathcal{K}, \tag{10}$$

$$\sum_{i\in\mathcal{N}} y_{ik} = u_k, \qquad \forall k \in \mathcal{K}, \tag{11}$$

$$l_{ijk} \geq \bar{l}_{ij} - M(2 - y_{ik} - z_{jk}), \qquad \forall i, j \in \mathcal{N},\ k \in \mathcal{K}, \tag{12}$$

$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} l_{ijk} \leq \bar{l}_k, \qquad \forall k \in \mathcal{K}. \tag{13}$$

Constraints (10) and (11) ensure that each service area contains exactly one centre. Constraint (12) defines the travel distance between centre i and station j. In detail, l_{ijk} equals $\bar{l}_{ij}$ if both stations i and j are in service area k and station i is a centre; it is 0 otherwise. Constraint (13) restricts the total travel distance of vehicle k, which cannot exceed a given threshold. This threshold can be estimated by the average speed of vehicles, the service time at each station, and the period length. Note that the total distance on the left-hand side of constraint (13) can be considered an upper bound of the travel distance of the vehicles because (i) it may not be necessary to visit all the stations to satisfy the real demand, and (ii) the vehicle can rebalance bikes between two stations directly rather than visit the centre midway.

The fourth group of constraints concerns the relationships among the number of bikes at each station during period t, the total demand, the covered demand and the rebalancing operations.

$$r_{ijt} \leq Q \sum_{k\in\mathcal{K}} \lambda_{ijk}, \qquad \forall i, j \in \mathcal{N},\ t \in \mathcal{T}. \tag{14}$$

$$v_{it} = v_{it-1} + \sum_{j\in\mathcal{N}} (r_{ji,t-1} - r_{ij,t-1} + w_{ji,t-1} - w_{ij,t-1}), \qquad \forall i \in \mathcal{N},\ t \in \mathcal{T}/\{0\}, \tag{15}$$

$$w_{ijt} \leq \tilde{d}_{ijt}, \qquad \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, \tag{16}$$

$$\sum_{j\in\mathcal{N}} w_{ijt} \leq v_{it} + \sum_{j\in\mathcal{N}} r_{jit} - \sum_{j\in\mathcal{N}} r_{ijt}, \qquad \forall i \in \mathcal{N},\ t \in \mathcal{T}. \tag{17}$$

Constraint (14) restricts the number of bikes rebalanced from station i to j only if these two stations are in the same service area; the number of rebalanced bikes cannot exceed the vehicle's capacity. Constraint (15) defines the flow conservation of each station. That is, the number of available bikes in period t equals the sum of the number of bikes rebalanced and the net inflow of bikes from the fulfilled demand. For each origin-destination (OD) pair from zone i to j, constraint (16) means the covered demand cannot exceed the total demand, while constraint (17) means that the covered demand is no larger than the number of available bikes after the rebalancing operation.

The objective function of our model (18) aims to maximize the bike system's revenue minus the total cost of bike rebalancing and the fixed operation cost of rebalancing the vehicles.

$$\max\ f \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} w_{ijt} - \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{1ij} r_{ijt} - c_2 \sum_{k\in\mathcal{K}} u_k \tag{18}$$

$$\text{s.t.}\ (1) - (17)$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\lambda} \in \{0, 1\}, \tag{19}$$

$$\boldsymbol{v}^0, \boldsymbol{l} \geq 0, \tag{20}$$

$$\boldsymbol{v}^{-0}, \boldsymbol{r}, \boldsymbol{w} \geq 0. \tag{21}$$

Constraints (19)–(21) are the variable definition constraints, where $\boldsymbol{v}^0 = (v_{i0})_{i\in\mathcal{N}}$ and $\boldsymbol{v}^{-0} = (v_{i,t\neq 0})_{i\in\mathcal{N}}$ are the initial inventory and state inventory variables during the decision periods, respectively.

The constraints are divided into two constraint sets corresponding to the two stages of our model. The set $\mathcal{X}_1$ contains constraints (1)–(2), (3) for $t = 0$, (4)–(13), and (19)–(20), which correspond to the strategic decision process and determine the station positions, initial bike inventories, and service areas. We denote the set $\mathcal{X}_1$ as the first-stage constraint set. Constraints (3) for $t > 0$, (14)–(17), and (21) compose the second-stage constraint set $\mathcal{X}_2(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{v}^0, \tilde{\boldsymbol{d}})$, which defines the fleet operation over the decision periods, including the fleet rebalancing operation, covered demand, and inventory flow conservation. Note that although our model involves a large number of parameters and variables, it is easy to replicate and use in practice. The potential demand can be estimated by GIS-based methodology (Garcìa-Palomares et al., 2012) and real taxi trajectory data (Park and

Sohn, 2017). In our study, we use open-access bike-sharing data to test our model. Other parameter values can be obtained in practice, such as the distance between two stations and the capacity and speed of each vehicle. In section 5, we describe the procedure that generates the inputs of our model.

3.3. Uncertainty set

To address the demand uncertainty, robust optimization and stochastic programming are two common approaches used in strategic and operational decisions. Stochastic programming assumes that the exact distribution of random parameters (demand in this study) is known, which can be estimated based on a large amount of reliable data. In practice, the distribution of the true demand is ambiguous for bike-sharing firms. For example, for a given station, since users cannot access the system if there are no available bikes, the bike-sharing firm can only observe the covered flows rather than the real demand. Moreover, any demand estimation based on historical data is also subject to uncertainty, particularly when the size of the available data set is limited. Due to the limited data, we adopt the robust optimization approach, which describes random demand via the uncertainty set.

We assume that the demand lies in an uncertainty set $\mathcal{U}$, which includes two parts. First, the uncertain OD demand $\tilde{\boldsymbol{d}}$ is assumed to be located in an interval; that is, $\tilde{\boldsymbol{d}} \in [\underline{\boldsymbol{d}}, \overline{\boldsymbol{d}}]$. The lower and upper bounds of this interval can be estimated by the minimum and maximum values of historical data. Second, considering that there are potential correlations between partial OD pairs, i.e., not all OD pairs can reach the lowest level simultaneously, we take the spatial correlations of OD pairs into account to avoid over conservation. Hence, $\mathcal{U}$ is defined as follows:

$$\mathcal{U} = \{\boldsymbol{d} \in [\underline{\boldsymbol{d}}, \overline{\boldsymbol{d}}], \sum_{(i,j)\in\mathcal{OD}_{st}} d_{ijt} \geq \Gamma_{st}, \forall s \in \mathcal{S}, t \in \mathcal{T}\}, \tag{22}$$

The second constraint in set (22) implies that the OD demands in a subset of OD pairs $\mathcal{OD}_{st}$ in each period have strong correlations and will not be less than a given threshold Γ_{st}. Such constraint is realistic; for example, the stations close by will be more correlated with demand. We can construct these subsets based on the demand covariance based on limited data. Γ_{st} can be set by decision makers based on their conservation levels. Note that this polyhedral form of uncertainty set (22) is similar to the budget uncertainty set presented by Bertsimas and Sim (2004), and it can be degraded into the box set when $\Gamma_{st} = \sum_{(i,j)\in\mathcal{OD}_{st}} \underline{d}_{ijt}$.

Our objective function can now be minimized by solving the following problem, where the variable Y is the revenue of the second-stage decision.

$$\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{v}^0,\boldsymbol{l}\in\mathcal{X}_1} Y - c_2 \sum_{k\in\mathcal{K}} u_k \tag{23}$$

$$\text{s.t.} \quad Y \leq \min_{\boldsymbol{d}\in\mathcal{U}} \max_{\boldsymbol{v}^{-0},\boldsymbol{r},\boldsymbol{w}\in\mathcal{X}_2(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{v}^0,\tilde{\boldsymbol{d}})} f \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} w_{ijt} - \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} c_{1ij} r_{ijt} \tag{24}$$

The robust model maximizes the worst-case revenue under low demand level. The worst-case revenue may not make sense to the planning makers who are more concerned about expected revenue over all possible demand realizations. However, the robust approach is already used to optimize many shared mobility systems when making strategic and operational decisions, such as ride-sharing (Hao et al., 2020), bike-sharing (He et al., 2017) and car-sharing systems (He et al., 2020). Furthermore, the station location problem in our study is a facility location problem. Some relevant studies also adopt robust and distributionally robust approaches to minimize the worst-case expected transshipment cost with satisfied model performance (Gülpınar et al., 2013; Basciftci et al., 2019; Saif and Delage, 2020). In section 5, the results show that the out-of-sample performance (i.e., mean revenue, worst-case revenue and standard deviation of revenue) of our robust model is much better than that of the stochastic model. Furthermore, since we also present a deterministic model above, the decision makers can choose one of these two models according to the actual situation.

4. Solution approach

Since constraint (24) is non-convex, it cannot be solved by state-of-the-art commercial solvers. To solve a program with a non-convex min-max form, duality theory is widely applied to reformulate the innermost maximum problem. By strong duality, the reformulation of the right-hand side of (24) is as follows:

$$\begin{aligned}
\min_{\boldsymbol{d}\in\mathcal{U}} \min_{\boldsymbol{\pi}^1,\boldsymbol{\pi}^2,\boldsymbol{\pi}^3,\boldsymbol{\pi}^4,\boldsymbol{\pi}^5\geq 0} \quad & \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} \tilde{d}_{ijt}\pi^4_{ijt} + \sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}/\{0\}} Cx_i\pi^1_{it} + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} Q\lambda_{ijk}\pi^2_{ijt} + \sum_{i\in\mathcal{N}} v_{i0}(\pi^3_{i0}+\pi^5_{i0}) && (25)\\
\text{s.t.}\quad & \pi^1_{it} + \pi^3_{it-1} - \kappa_t\pi^3_{it} - \pi^5_{it} \geq 0, \qquad \forall i\in\mathcal{N},\ t\in\mathcal{T}, && (26)\\
& \kappa_t(\pi^3_{it} - \pi^3_{jt}) + \pi^4_{ijt} + \pi^5_{it} \geq f, \qquad \forall i,j\in\mathcal{N},\ t\in\mathcal{T}, && (27)\\
& \pi^2_{ijt} + \kappa_t\pi^3_{it} - \kappa_t\pi^3_{jt} + \pi^5_{it} - \pi^5_{jt} \geq -c_{1ij}, \qquad \forall i,j\in\mathcal{N},\ t\in\mathcal{T}, && (28)
\end{aligned}$$

where $\kappa_t = 1$ if $t < |\mathcal{T}|$ and is 0 otherwise. $\boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \boldsymbol{\pi}^3, \boldsymbol{\pi}^4$ and $\boldsymbol{\pi}^5$ are the dual variables corresponding to constraints (3) and (14)–(17). We denote the vector of all dual variables as $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \boldsymbol{\pi}^3, \boldsymbol{\pi}^4, \boldsymbol{\pi}^5)$, and the dual polyhedron Π is defined as below:

$$\Pi = \{\boldsymbol{\pi} \mid \boldsymbol{\pi} \geq 0 \quad \text{satisfying constraints (26)-(28)}\}, \tag{29}$$

Based on the reformulation above, constraint (24) is now equivalent to

$$Y \leq \sum_{i\in\mathcal{N}}\Big(\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} \tilde{d}_{ijt}\pi^4_{ijt} + \sum_{t\in\mathcal{T}/\{0\}} Cx_i\pi^1_{it} + \sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} Q\lambda_{ijk}\pi^2_{ijt} + v_{i0}(\pi^3_{i0}+\pi^5_{i0})\Big), \qquad \forall\boldsymbol{\pi}\in\Pi,\ \tilde{\boldsymbol{d}}\in\boldsymbol{\mathcal{U}}. \tag{30}$$

Since constraint (30) is enforced over all representations of $\tilde{\boldsymbol{d}}$, the primal program contains an infinite number of constraints. This program (25)–(28) is a disjoint bilinear program because of the term $\sum_{i\in\mathcal{N}}\sum_{t\in\mathcal{T}} d_{it}\pi^4_{it}$. Hence, we cannot obtain a deterministic equivalent model, and we present the following two approaches.

4.1. Scenario-based model

A stochastic program is a common method to solve optimization problems under uncertainty. The basic idea of a two-stage stochastic program is to generate a set of scenarios to represent the possible outcome in advance. In our study, these are the various demand outcomes (Birge and Louveaux (2011)). In detail, we relax constraint (30) by sampling a set of scenarios Ω so that a near-optimal solution of program (23)-(24) can be obtained. That is, the primal model contains only a finite number of constraints; we call it a scenario-based model (SBM), and it is formulated as follows:

$$\begin{aligned}
\text{[SBM]} \quad \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{v}^0,\boldsymbol{l}\in\mathcal{X}_1} \quad & Y - c_2\sum_{k\in\mathcal{K}} u_k && (31)\\
\text{s.t.}\quad & v^\omega_{i0} = v_{i0}, \qquad \forall i\in\mathcal{N},\ \omega\in\Omega, && (32)\\
& \boldsymbol{v}^{-0\omega}, \boldsymbol{r}^\omega, \boldsymbol{w}^\omega \in \mathcal{X}_2(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{v}^{0\omega}, \boldsymbol{d}^\omega), \qquad \forall\omega\in\Omega, && (33)\\
& Y \leq f\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} w^\omega_{ijt} - \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{1ij}r^\omega_{ijt}, \qquad \forall\omega\in\Omega. && (34)
\end{aligned}$$

When Y is no large than the expected revenues over all scenarios in constraint (34), the SBM is a two-stage stochastic model. For a stochastic program, as the number of scenarios increases, the expected objective value may converge to the global optimal value. However, the worst-case objective function value is not theoretically able to converge, and this is our case. Therefore, the SBM will be contrasted with the row generation approach below. In addition, solving a stochastic program becomes more time-consuming as the number of scenarios increases. Hence, in the next section, we use a row generation approach to solve our model to optimality.

4.2. Row generation approach

Unlike the SBM, which uses a set of scenarios generated in advance, we generate each scenario and its associated constraint iteratively in the row generation approach. This idea is similar to Benders decomposition, which divides the original model into a *master problem* (MP) and a *subproblem* (SP). We first present the formulation of the MP and SP and then design the row generation approach to solve our original robust model.

The formulation of the MP is shown below.

$$\text{[MP]} \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{v}^0,\boldsymbol{l}\in\mathcal{X}_1} Y - c_2 \sum_{k\in\mathcal{K}} u_k \tag{35}$$

$$\text{s.t. } Y \leq \sum_{i\in\mathcal{N}} \Big(\sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \hat{d}^{\omega}_{ijt} \hat{\pi}^{4\omega}_{ijt} + \sum_{t\in\mathcal{T}/\{0\}} C x_i \hat{\pi}^{1\omega}_{it} + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} Q\lambda_{ijk} \hat{\pi}^{2\omega}_{ijt} + v_{i0}(\hat{\pi}^{3\omega}_{i0} + \hat{\pi}^{5\omega}_{i0}) \Big), \forall \omega \in \Omega. \tag{36}$$

Initially, we solve model (35) without the constraint set (36). Consequently, we obtain the optimal solutions $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{\lambda}}$, and $\hat{\boldsymbol{v}}^0$ for the MP. Based on the obtained solution, we generate one or several constraints in set (30) corresponding to various values of $\boldsymbol{d}$ and $\boldsymbol{\pi}$ and add them to the MP. We repeat the above processes until no valid constraints can be generated.

To identify a new constraint, similar to Benders decomposition, a subproblem should be created and used to generate valid cuts for the master problem. We employ program (25)–(28) with the incumbent solutions $\hat{\boldsymbol{x}}, \hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{v}}^0$ as the SP.

As mentioned above, the inner program of (25) is a disjoint bilinear program. Proposition 1 transforms this program and obtains an equivalent MILP model, shown below.

Proposition 1. *The SP can be reformulated as follows:*

$$\min \sum_{i\in\mathcal{N}} \Big(\sum_{t\in\mathcal{T}/\{0\}} C\hat{x}_i \pi^1_{it} + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} Q\hat{\lambda}_{ijk} \pi^2_{ijt} + \hat{v}_{i0}(\pi^3_{i0} + \pi^5_{i0}) + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} (\underline{d}_{ijt} \underline{u}_{ijt} - \overline{d}_{ijt} \overline{u}_{ijt}) \Big) + \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \Gamma_{st} u_{st}$$

$$\begin{aligned}
\textit{s.t.}\ & \boldsymbol{\pi} \in \Pi, \\
& \pi^4_{ijt} - \sum_{s:(i,j)\in\mathcal{OD}_{st}} u_{st} + \overline{u}_{ijt} - \underline{u}_{ijt} = 0, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (37) \\
& \sum_{(i,j)\in\mathcal{OD}_{st}} d_{ijt} \geq \Gamma_{st}, && \forall s \in \mathcal{S},\ t \in \mathcal{T}, && (38) \\
& -d_{ijt} \geq -\overline{d}_{ijt}, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (39) \\
& d_{ijt} \geq \underline{d}_{ijt}, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (40) \\
& u_{st}\Big(\sum_{(i,j)\in\mathcal{OD}_{st}} d_{ijt} - \Gamma_{st} \Big) = 0, && \forall s \in \mathcal{S},\ t \in \mathcal{T}, && (41) \\
& \overline{u}_{ijt}(-d_{ijt} + \overline{d}_{ijt}) = 0, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (42) \\
& \underline{u}_{ijt}(d_{ijt} - \underline{d}_{ijt}) = 0, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (43) \\
& \boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \boldsymbol{\pi}^3, \boldsymbol{\pi}^4, \boldsymbol{\pi}^5, \boldsymbol{d}, \boldsymbol{u}, \overline{\boldsymbol{u}}, \underline{\boldsymbol{u}} \geq 0. && && (44)
\end{aligned}$$

Proof. By strong duality theory, we reformulate the subproblem (25)–(28) as follows:

$$\min \Big\{ \sum_{i\in\mathcal{N}} \Big(\sum_{t\in\mathcal{T}/\{0\}} C\hat{x}_i \pi^1_{it} + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} Q\hat{\lambda}_{ijk} \pi^2_{ijt} + \hat{v}_{i0}(\pi^3_{i0} + \pi^5_{i0}) \Big) + \max\{ \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \Gamma_{st} u_{st} + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} (\underline{d}_{ijt} \underline{u}_{ijt} - \overline{d}_{ijt} \overline{u}_{ijt}) \} \Big\}$$

$$\begin{aligned}
\text{s.t. } & \boldsymbol{\pi} \in \Pi,\ (37), \\
& \boldsymbol{u}, \overline{\boldsymbol{u}}, \underline{\boldsymbol{u}} \geq 0.
\end{aligned}$$

This formulation is a linear min-max model and can be further regarded as a bilevel problem:

$$\min_{\boldsymbol{\pi}\geq 0} \sum_{i\in\mathcal{N}} \Big(\sum_{t\in\mathcal{T}/\{0\}} C\hat{x}_i \pi^1_{it} + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{k\in\mathcal{K}} Q\hat{\lambda}_{ijk} \pi^2_{ijt} + \hat{v}_{i0}(\pi^3_{i0} + \pi^5_{i0}) + \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} (\underline{d}_{ijt} \underline{u}_{ijt} - \overline{d}_{ijt} \overline{u}_{ijt}) \Big) + \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \Gamma_{st} u_{st} \tag{45}$$

$$\text{s.t. } \boldsymbol{u}, \overline{\boldsymbol{u}}, \underline{\boldsymbol{u}} \in \arg\max \Big\{ \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \Gamma_{st} u_{st} + \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} (\underline{d}_{ijt} \underline{u}_{ijt} - \overline{d}_{ijt} \overline{u}_{ijt}) \Big|$$

$$\boldsymbol{\pi}, \boldsymbol{u}, \overline{\boldsymbol{u}}, \underline{\boldsymbol{u}} \geq 0 \text{ and satisfy the constraints } \Pi, (37)\Big\}.$$

Then, adopting the Karush-Kuhn-Tucker (KKT) condition in the lower-level problem, we can obtain the formulation shown in Proposition 1. □

Equations (26)–(28) are the constraints corresponding to the variables of the outer problem. Constraints (37) and (38)–(40) guarantee primal and dual feasibility, and constraints (41)–(43) guarantee the KKT conditions, which can also be represented as linear constraints using binary variables and the big-M method. Constraints (41)–(43) can be rewritten as follows:

$$\begin{aligned}
&\sum_{(i,j)\in\mathcal{OD}_{st}} d_{ijt} - \Gamma_{st} \leq M\mu_{st}, && \forall s \in \mathcal{S},\ t \in \mathcal{T}, && (46)\\
&u_{st} \leq M(1-\mu_{st}), && \forall s \in \mathcal{S},\ t \in \mathcal{T}, && (47)\\
&-d_{ijt} + \overline{d}_{ijt} \leq M\overline{\mu}_{ijt}, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (48)\\
&\overline{u}_{ijt} \leq M(1-\overline{\mu}_{ijt}), && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (49)\\
&d_{ijt} - \underline{d}_{ijt} \leq M\underline{\mu}_{ijt}, && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (50)\\
&\underline{u}_{ijt} \leq M(1-\underline{\mu}_{ijt}), && \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, && (51)\\
&\boldsymbol{\mu}, \overline{\boldsymbol{\mu}}, \underline{\boldsymbol{\mu}} \in \{0, 1\}. && && (52)
\end{aligned}$$

After constructing the MILP form of the subproblem, constraint (36) can be generated easily. Let Z_{SP} be the optimal value of the subproblem, and let $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{d}}$ be the corresponding optimal solutions. If $Y \leq Z_{SP}$, then constraint (36) is satisfied for all $\boldsymbol{d} \in \mathcal{U}$, so the incumbent solution is also the optimal solution of primal program (23). Otherwise, a new constraint (36) is added to the master problem. The master problem is solved again to obtain a new optimal solution. The framework of the row generation approach is shown in Algorithm 1.

Algorithm 1: Row generation approach

```
1: Initialize $\omega = 0$.
2: Solve the MP, and obtain the optimal value $Z_{MP}$ with solutions $\hat{\boldsymbol{x}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{v}}^0$ and $\hat{Y}$.
3: Solve the SP with fixed $\hat{\boldsymbol{x}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{v}}^0$ to obtain the optimal value $Z_{SP}$ and corresponding solutions $\hat{\boldsymbol{\pi}}^\omega, \hat{\boldsymbol{d}}^\omega$.
4: If $\hat{Y} \leq Z_{SP}$, stop the algorithm; $Z_{MP}$ is the optimal value of the original program. Otherwise, add constraint
   (36) to the MP, set $\omega = \omega + 1$, and return to step 2.
```

Note that although ω in a SBM is the index of the scenarios, it is also used as the iteration index in algorithm 1. These two uses are consistent because in each iteration of algorithm 1, only one valid cut is added to the MP, which uniquely corresponds to a scenario. As mentioned above, constraint (30) is enforced over all possible values of $\boldsymbol{d} \in \mathcal{U}$. Thus, the question of whether the row generation algorithm can converge within finite steps can be answered. Theorem 2 ensures convergence and is proven below.

Proposition 2. *The row generation approach can converge within finite steps under the uncertainty set $\mathcal{U}$.*

Proof. Finite convergence can be guaranteed if each constraint (36) is associated with a vertex of the polyhedron defined by $\mathcal{U}$. Consider the minimization problem (25)–(28), which is the prototype of the SP. Assume that $\boldsymbol{d}$ and $\boldsymbol{\pi}$ are the optimal solution with the optimal objective value $Y(\boldsymbol{d}, \boldsymbol{\pi})$ and that $\boldsymbol{d}$ is not associated with a vertex. Hence, we have $Y(\boldsymbol{d}, \boldsymbol{\pi}) \geq Y(\boldsymbol{d}', \boldsymbol{\pi}), \forall \boldsymbol{d}' \in \mathcal{U}$. Then, fix $\boldsymbol{\pi}$, and the problem is now a linear program, which can be solved by a simplex algorithm. Hence, the associated solution $\boldsymbol{d}'$ must correspond to a vertex. The optimal objective value is denoted as $Y(\boldsymbol{d}', \boldsymbol{\pi})$ and is no larger than $Y(\boldsymbol{d}, \boldsymbol{\pi})$. Hence, $Y(\boldsymbol{d}, \boldsymbol{\pi}) = Y(\boldsymbol{d}', \boldsymbol{\pi})$, so each objective value corresponds to a vertex of the polyhedron. Clearly, the number of vertices is finite, so constraint (36) is enumerable and the algorithm can converge finitely. □

4.3. Row generation enhancement

To handle the large-scale problem, some remarks on implementing algorithm 1 are provided. First, we can further tighten our model by rewriting, adding, and deleting some constraints. Observe that the value of d_{ijt}

cannot reach its lower and upper bounds simultaneously. Thus, at least one of $\overline{\mu}_{ijt}$ and $\underline{\mu}_{ijt}$ is larger than one. We can tighten the above model by adding the following constraint:

$$\overline{\mu}_{ijt} + \underline{\mu}_{ijt} \geq 1, \qquad \forall i, j \in \mathcal{N}, t \in \mathcal{T}. \tag{53}$$

Furthermore, the big-M in some constraints can be replaced by an explicit value. For example, M in constraints (48) and (50) can be set to Δd_{it} based on the box region of d_{ijt}, where $\Delta d_{ijt} = \overline{d}_{ijt} - \underline{d}_{ijt}$. Theorem 2 shows that the optimal value of $\boldsymbol{d}$ can only correspond to one vertex of the polyhedron defined by $\boldsymbol{\mathcal{U}}$. Thus, inequality (38) can become an equation, and constraint (41), or constraints (46) and (47), can be removed.

Second, to generate a constraint in step (4) of algorithm 1, it is not necessary to solve the optimization problem optimally. Instead, the algorithm can add a cut whenever the MILP solver finds a feasible solution for the SP such that the objective function is strictly less than $\hat{Y}$. Based on the associated solution, we can generate a valid constraint in each iteration and save computational time. Furthermore, once a feasible solution is found, we can use it as an initial point to solve the bilinear program (45) to a local optimum by a greedy heuristic method. That is, we fix either $\boldsymbol{d}$ or $\boldsymbol{\pi}$ and optimize over the other iteratively. Since this method involves only linear optimization, it can be solved quickly. In our implementation of this method, the last iteration is always an optimization over $\boldsymbol{d}$ with fixed $\boldsymbol{\pi}$. Hence, this ensures that algorithm 1 can always generate constraint (36) at each iteration if the termination condition is unmet.

Third, similar to other row-generation-based approaches, algorithm 1 also faces the common problem that the lower bound converges slowly. To address this problem, we employ the idea of partial Benders decomposition (Crainic et al., 2016). In detail, we generate a sample of $\tilde{\boldsymbol{d}}$ and add the relevant constraints (32)-(34) into the master problem. We define this problem as a partial master problem (PMP), which is now rewritten as:

$$[\text{PMP}] \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{v}^0,\boldsymbol{l}\in\mathcal{X}_1} Y - c_2 \sum_{k\in\mathcal{K}} u_k \tag{54}$$

$$\text{s.t. Constraints (32)} - \text{(34)},\ \text{(36)}$$

A PMP can be considered as a SBM with constraint (36). A large number of samples added to the PMP model cannot guarantee a great lower bound but may result in additional computational time for the PMP model. We suggest adding a few samples corresponding to the vertexes of $\boldsymbol{\mathcal{U}}$; that is, the samples can be the solutions of the following model:

$$d_{ijt} \in \arg\min \left\{ \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} q_{ijt} d_{ijt} | \ \boldsymbol{d} \in \boldsymbol{\mathcal{U}} \right\}, \tag{55}$$

where $\boldsymbol{q}$ is a random weight vector.

5. Numerical examples

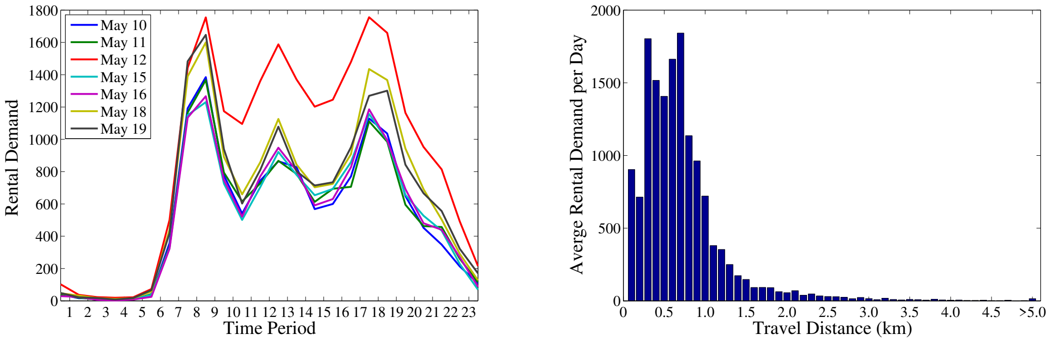
In this section, we test our robust model and algorithm using an open data set. The data and parameter setting descriptions are shown in section 5.1. To demonstrate the efficiency of our algorithm, section 5.2 compares the algorithm with the SBM based on a small network, different parameter settings, and scenarios associated with random demand. For a real-size network, section 5.3 investigates the performance of our robust model from the perspective of the mean and worst-case performance with several parameter settings. All the numerical experiments are implemented in Python 2.7 and CPLEX 12.7.1 on a machine with a 64-bit Windows 7 operating system, eight Intel Core i7 CPUs at 3.6 GHz, and 16 GB RAM.

5.1. Description of the open data and parameter settings

This section presents the parameter settings for both a small network and a large real-size network. Since the small network is generated based on the large network, we first describe the latter and then the former in sections 5.1.1 and 5.1.2, respectively. Finally, section 5.1.3 gives the computational parameters of these two networks.

5.1.1. Input settings for the real-size network

The data in our case study are from Mobike in Dongcheng District, Beijing, China (https://biendata.com/competition/mobike/). Dongcheng District is one of the central districts between east longitudes 116°22′17″ and 116°26′46″ and north latitudes 39°51′26″ and 39°58′22″ and has an area of 41.84 km^2. In detail, these data include actual trip information with seven fields (i.e., (1) order number, (2) user ID, (3) bike ID, (4) bike type, (5) start time of the trip, (6) longitude and latitude of the trip origin, and (7) longitude and latitude of the trip destination) from May 10 to May 24, 2017, except for May 14 and 17. In our study, only the last three fields are useful in constructing the OD distribution. We further removed the weekend data (May 13, 20, and 21), rainy day data (May 22) and days with too much data loss (May 23 and 24) to reduce data bias. Hence, there are only seven days of data, with 103,730 records. Fig. 3(a) shows the demand per hour for the seven days. A large proportion of the demand is from 7 am to 11 pm; thus, we omit other time periods. Since the size of the data set is too small to precisely estimate the probability distribution of the demand, a robust model is appropriate. To handle the problem, we show the parameter settings based on historical data below.



(a)

(b)

Figure 3: Demand distribution for each period and the trip distance distribution for all OD pairs

- *Setting the period*

Since the records do not contain trip end time information, the travel time is estimated based on the average bike-riding speed and the Euclidean distance between the origin and destination. Jensen et al. (2010) suggests that the average speed of bike-riding is 13.5 to 15 km/h, and Mobike states that the average speed in each city is different, ranging between 6.5km/h and 10km/h. Fig. 3(b) shows the Euclidean distance between the origin and destination of all trips, most of which are in the range of 0.2km to 1.5km. The travel time for most trips is no more than one hour. Note that both in this work and a large number of existing studies (Nair and Miller-Hooks, 2014; Frade and Ribeiro, 2015; Lu et al., 2017), an implicit assumption is that each bike can only be used one time in a period. Hence, short period length can estimate the revenue more exactly.

However, it does not imply that the period length should be as short as possible. A short period length results in a large number of decision periods and increases the computational time of our model. Based our interview with the staff of a bike-sharing firm, many factors influence the period length, such as the operation speed of bike loading/unloading, time spent on broken bike collection, and bike cleaning. Therefore, one period of a rebalancing operation cannot be completed very fast, for example, one hour. Another interview shows that each station should be visited by rebalancing vehicles at least three times per workday. Thus, we suggest that the period length should vary from four to six hours. From a practical perspective, we recommend that the period length and the related number of periods be determined based on realistic considerations, as well as computational requirements.

Table 2 shows the computational time (seconds) and gaps (in brackets, %) of the deterministic model under different numbers of periods. The maximum travel distance of rebalancing vehicles is dependent on the period length, which is essentially the same as the number of periods; that is, $\bar{l}_k = 10, 20$ and 40, if $|\mathcal{T}| = 16, 8$ and 4, respectively. The other parameter settings are discussed in the following section. The model with 16 periods results in suboptimal solutions with large gaps. Since the number of periods does not impact the procedure of the algorithm, the period length in our study is set to two hours, and the total number of periods is $|\mathcal{T}| = 8$.

Table 2: Computational time and gaps (%) for the deterministic model with different numbers of periods

$\lvert\mathcal{T}\rvert$	$B_1 = 20$				$B_1 = 40$			
	$B_2 = 250$	$B_2 = 500$	$B_2 = 750$	$B_2 = 1000$	$B_2 = 250$	$B_2 = 500$	$B_2 = 750$	$B_2 = 1000$
4	2.78	2.67	2.67	2.86	2.66	2.64	2.8	2.7
8	5.02	4.58	7200 (3.29)	7200 (2.79)	10.53	9.36	8.81	9.56
16	7200 (1.30)	7200 (22.94)	7200 (34.90)	7200 (33.77)	28.41	7200 (1.00)	7200 (4.38)	7200 (9.02)

- *Setting the zone generation, OD pairs, and trip distance*

Recall that each zone is actually a candidate station location. We generate a set of zones $\mathcal{N}$ (see the circles in Fig. 2) to capture potential bike locations. A mixed integer program model is designed to generate all candidate bike locations. The detailed model is shown in appendix A. For the real-size network, we set $|\mathcal{N}| = 55$, with a total of 992 OD pairs in this region. The associated demand of OD pair (i, j) in period t and day h is denoted as d_{ijth}. The service area is composed of several zones. Each service area contains a centre station. We employ the Euclidean distance between the centres of service areas i and j to describe the corresponding distance $\bar{l}_{ij}$ (km). A large number of OD pairs leads to a computational challenge. Observe that a small number of OD pairs occupy a large proportion of the flows. We aim to use an auxiliary model under different numbers of OD pairs. Thus, we can determine a proper size for the OD pair set based on the trade-off between the computational efficiency and objective value. Specifically, we call this auxiliary model the deterministic model. The deterministic model has the same model structure as our two-stage robust model in section 3.2, but $\tilde{d}_{ijt}$, the right-hand side of constraint (16), is replaced by the mean of $\underline{d}_{ijt}$ and $\overline{d}_{ijt}$, which is defined as follows:

$$\overline{d}_{ijt} = \max_h\{d_{ijth}\},\ \forall i, j \in \mathcal{N},\ t \in \mathcal{T}, \tag{56}$$

$$\underline{d}_{ijt} = \min_h\{d_{ijth}\},\ \forall i, j \in \mathcal{N},\ t \in \mathcal{T}. \tag{57}$$

Various sizes of OD pair sets with the largest flows are another input. Table 3 and Fig. 4 show the computational time and optimal objective value of each instance associated with the deterministic model.

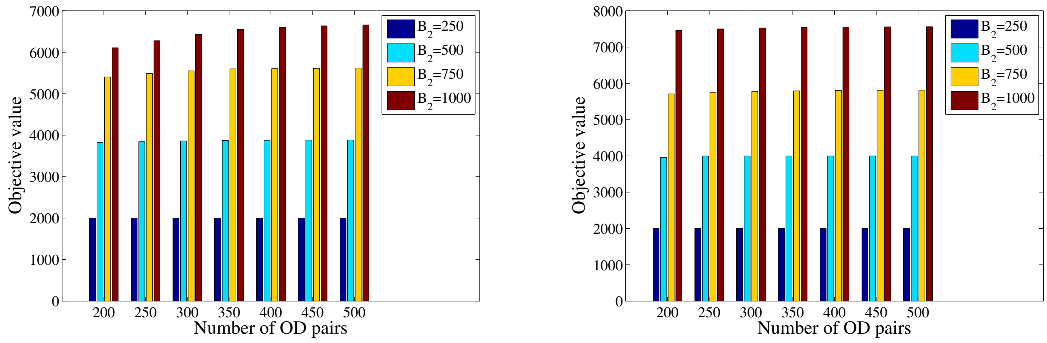
Table 3: Computational time for the deterministic model with various parameters

OD	$B_1 = 20$				$B_1 = 40$			
	$B_2 = 250$	$B_2 = 500$	$B_2 = 750$	$B_2 = 1000$	$B_2 = 250$	$B_2 = 500$	$B_2 = 750$	$B_2 = 1000$
100	4.23	2157.35	7200	7200	2.98	58.59	188.56	7200
150	3.32	32.78	7200	7200	3.32	137.73	2088.18	7200
200	3.2	14.68	7200	7200	3.4	1238.54	147.23	14.32
250	3.14	27.67	7200	7200	3.37	5.51	8.38	7.63
300	3.6	8.19	7200	7200	3.45	3.43	8.13	13.12
350	3.56	8.02	1652.47	7200	3.59	3.48	7.55	11.14
400	3.54	8.49	3489.21	7200	3.37	3.42	8.07	10.42
450	3.79	8.05	3359.22	7200	3.45	3.48	10.34	8.28
500	3.32	11.48	1743.72	7200	3.48	3.46	7.18	7.46

Although several instances do not converge within two hours, the maximum gap is 1.64%, and the solutions are still acceptable. From the perspective of computational time, the interesting result is that a smaller number of OD pairs does not always mean less computational time. A possible reason is that the demand in different zones is dispersed so that the model has difficulty deciding which zones should be placed at a station. As the size of the OD set increases, some zones may have greater demand than others. Zones with greater demand may have a higher likelihood of becoming locations. Further increasing the number of OD pairs does not reduce the computational time because of the existence of a substantial number of binary variables. Thus, a setting of slightly more than 300 OD pairs is appropriate. The objective value increases in the number of OD pairs with diminishing marginal returns. Comparing the objective values of 350 to 500 OD pairs, the maximal revenue increment over these instances is only 5.16%. Hence, by jointly taking the computational effectiveness and accuracy into account, we choose 350 OD pairs with the largest OD flows in our numerical experiments.

- *Setting the budget, revenue, cost, maximal travel distance, and vehicle capacity*

We set the revenue per trip $f = 1$ following the actual price used by Mobike, and the rebalancing cost per



(a)

(b)

Figure 4: Objective value associated with various OD pairs and B_2 under (a) $B_1 = 20$ and (b) $B_1 = 40$

bike c_{1ij} is proportional to the travel distance $\bar{l}_{ij}$. We set $c_{1ij} = 0.01\bar{l}_{ij}$. A relatively small value for the fixed cost of each rebalancing vehicle c_2 is 5. The number of rebalancing vehicles K and their capacities Q are 5 and 20, respectively. The maximal travel distance of vehicle $\bar{l}_k$ is set to 20 km. Since each arc between a centre and a station should be visited twice to ensure the connection of each route, this implies that each rebalancing vehicle can travel at most 40 km in two hours. This setting is realistic, and it makes the service time at stations sufficient. In addition, the vehicle speed in Beijing is usually less than $40km/h$.

To analyse the correlation between the budget and the model performance, we use various combinations of B_1, B_2, and C. The capacity of stations C lies in [50, 100], while the number of station locations ranges from 10 to 40. For the different numerical experiments, the number of bikes allocated B_2 belongs to various sets. The default set is $[250, 500, 750, 1000]$.

- *Setting the uncertainty set*

The constructed uncertainty set in this study includes three parameters, i.e., support set $\boldsymbol{d}$, OD set $\mathcal{S}$ and minimal demand threshold Γ_{st} (see Eqs. (22)). Here, we present a guideline to set these parameters including the method used in this study as well as other alternative methods.

First, we use the maximum and minimum sample values defined in Eqs. (56) and (57) to set the value of $\overline{\boldsymbol{d}}$ and $\underline{\boldsymbol{d}}$. Note that other approaches can be used to define the support set. In statistics and machine learning, bootstrap method is widely used to estimate the confidence interval of small dataset. This method does not depend on probability distribution of prior data, which is quite popular in some robust optimization literature (He et al., 2020).

Second, the demand OD pairs in a subset $s \in \mathcal{S}$ could be highly correlated. In practice, the demands of the OD pairs with the same origin are highly correlated over all periods. Hence, in our numerical study, we simplify the inequality in set (22) as $\sum_{j\in\mathcal{N}} d_{ijt} \geq \Gamma_{it}, \forall i \in \mathcal{N}, t \in \mathcal{T}$, i.e., $|\mathcal{S}| = |\mathcal{N}|$. In addition, other cluster approaches can also be used to classify OD pairs into $|\mathcal{S}|$ groups.

Third, because $|\mathcal{S}| = |\mathcal{N}|$, we let $\Gamma_{it} = \Gamma_{st} = (1-\alpha)\sum_{j\in\mathcal{N}} \overline{d}_{ijt} + \alpha \sum_{j\in\mathcal{N}} \underline{d}_{ijt}$. As mentioned above, the mean of $\underline{d}_{ijt}$ and $\overline{d}_{ijt}$ is employed as the input of the deterministic model. Hence, α should be larger than 0.5, and we assume it belongs to $[0.6, 0.7, 0.8, 0.9]$ to analyze the sensitivity of Γ_{it}.

In practice, decision maker can set different Γ_{st} to solve the model. Optimal value can be obtained based on simulation performance of solutions. Bootstrap is still an alternative method to estimate Γ_{st}. That is, we employ bootstrap method to generate several samples of the total demand in terms of the OD pairs belonging to set $\mathcal{OD}_{st}$, and then use the lower bound of these samples as Γ_{st}. Note that Γ_{st} describes the conservative level of decision. Decreasing the value of Γ_{st} can guarantee a higher probability that the worst-case scenario belonging to the uncertainty set (Bertsimas and Sim, 2004). However, the decision may be overconservative if Γ_{st} is quite small, since the worst-case scenario will not likely to be realized.

- *Model performance*

To examine the performance of the robust solution, we usually compare the robust model with the deterministic model (defined above) in robust optimization by a simulation-based method. This method consists of three parts, i.e., sample generation, revenue simulation and performance comparison.

Regarding sample generation, we assume that $\tilde{d}_{ijt}$ follows a two-point distribution. The sampled d_{ijt} is generated equal to either $\underline{d}_{ijt}$ or $\overline{d}_{ijt}$ according to the probabilities β and $(1-\beta)$, where $\beta \in [0.2, 0.4, 0.6, 0.8]$ is applied to simulate different real situations. For example, $\beta = 0.2$ may represent low demand under extreme weather conditions, while $\beta = 0.8$ is associated with high demand influenced by favourable weather, events, and festivals. The two-point distribution is widely used to perform out-of-sample evaluation of robust models (Bertsimas and Sim, 2004; Zhang et al., 2019), as well as other simple distributions, such as the normal distribution and uniform distribution (Ben-Tal et al., 2013; He et al., 2020; Hao et al., 2020). Note that since the true distribution is unknown, the correlation of random parameters is not considered in our sample generation process.

After generating samples, the first-stage solutions of the robust and deterministic models (see Table 1) are put into the simulation system. In detail, the first-stage solution consists of the station location, initial bike allocation, and service area design. Thereby, the total revenue corresponding to each scenario is obtained. We run 2,500 simulations for each β, which means that there are a total of 10,000 samples. Accordingly, we obtain 10,000 simulation solutions and associated revenues for our two-stage robust and deterministic models, respectively.

To compare the model performance, two metrics, namely, the worst-case performance and mean performance, are constructed based on the following equations (58) – (59).

$$\text{Worst-case performance} = \frac{\text{min obj. of robust solution - min obj. of deterministic solution}}{\text{min obj. of deterministic solution}} \times 100\%, \tag{58}$$

$$\text{Mean performance} = \frac{\text{mean obj. of deterministic solution - mean obj. of robust solution}}{\text{mean obj. of deterministic solution}} \times 100\%, \tag{59}$$

where *min obj. of robust solution* is the minimal revenue over all 10,000 simulations corresponding to the robust solution, while *mean obj. of robust solution* is the expected value of all simulated revenues corresponding to the robust solution. The meanings of *min obj. of deterministic solution* and *mean obj. of deterministic solution* follow similar logic.

These two metrics compare the simulated revenues between the robust and deterministic models from different perspectives. Since the robust model aims to optimize the worst-case revenue associated with a given uncertainty set, the high out-of-sample worst-case performance exhibits a high robustness and a satisfactory improvement in the lower bound of revenue. Several studies also adopt the out-of-sample worst-case objective value to describe the model performance (Ben-Tal et al., 2013) or use a similar metric such as the 95% (99%) Value at Risk (VaR, Jaillet et al., 2016; Qi et al. (2016)). The mean performance compares the average revenue between two models (Jaillet et al., 2016; Hao et al., 2020). The solution of the robust model with a low mean performance reflects high operational capability of a bike-sharing system. Note that other metrics, such as standard deviation (STD) of performance (Ben-Tal et al., 2004; Jaillet et al., 2016; Hao et al., 2020), price of robustness (Ben-Tal et al., 2004) and probability violation (Bertsimas and Sim, 2004) can also be used to evaluate model performance.

5.1.2. Input settings for the small network

Numerical experiments associated with a small network are constructed based on the real-size network above. The topological structure of this network is composed of 10 zones associated with the 10 largest OD pairs. The first two (or three or four) periods (i.e., $\mathcal{T} \in \{2, 3, 4\}$) are considered. Thus, there are a total of 56 OD pairs, and the part of this group with the largest flows is used to test the performance of our algorithm; that is, the number of OD pairs is set to 20, 40, and 56. Additionally, the numbers of stations and initial bikes allocated are 5 and 100, respectively. Regarding the parameters for rebalancing vehicles, we let $K = 1$ and $Q = 20$. The maximal travel distance of vehicles $\overline{l}_k$ is not restricted, which implies that constraints (5)–(9) are unnecessary.

For the inequality in the uncertainty set (22), we adopt a single inequality including all OD pairs; that is, $\sum_{(i,j)\in\mathcal{OD}} d_{ijt} \geq \Gamma_t, \forall t \in \mathcal{T}$. The value of α is 0.8. Other parameter settings are similar to the settings in section 5.1.1.

To verify the efficiency of the algorithm, we compare it to the SBM under a variety of scenarios. To generate each scenario, the worst-case situations correspond to the vertexes of polyhedron $\mathcal{U}$. The scenarios are the solutions obtained by model (55) with different values for $\boldsymbol{q}$.

In contrast to the two-stage stochastic program, which normally aims to optimize the expected objective function, the optimal objective value of our model does not converge as the number of scenarios increases. Hence, a large size of the scenario set Ω used in the scenario-based model is necessary. The size of this set varies from 100 to 500. However, compared with the total number of scenarios, these generated scenarios are still too small. Further increasing the number of scenarios will make the model computationally intractable. Therefore, to avoid the bias incurred by the scenario generation process, each scenario-based instance will be solved and repeatedly associated with five different sets of scenarios.

5.1.3. Computation-related parameters

For the small network, we do not limit the maximum computational time for scenario-based or robust models, whereas for the large real-size network, obtaining an optimal solution of the subproblem is time consuming. Based on the algorithmic process, we employ a feasible solution satisfying $\hat{Y} > Z_{SP}$ to construct a valid cut for the master problem. Thus, we set the maximum limited time for solving the subproblem to 100 seconds, which is enough to find a near-optimal solution with a small gap. To further improve performance, the optimal solution of the associated deterministic model is used as a warm start. Nevertheless, obtaining an optimal solution for such a large network is difficult. Instead, we aim to obtain the near-optimal solution that improves the worst-case performance of the deterministic model solution as much as possible. Thus, the maximum number of iterations is set to 100. The results demonstrate that the second-best approach can still obtain high-quality solutions and a trend of the worst-case performance corresponding to different parameter settings. For convenience, we summarize all the parameter settings in sections 5.1.1 to 5.1.3 in Table 4.

Table 4: Parameter settings for the robust model and the algorithm for two networks

Parameter	Small instance	Large instance
$\mathcal{N}$	10	55
Number of OD pairs	[20,40,56]	350
K	1	5
$\mathcal{T}$	[2,3,4]	8
Q	20	20
C	50	[50,100]
B_1	5	[10,20,30,40]
B_2	100	[250, 500, 750, 1000] in section 5.3.2 $[0.25B_1C, 0.5B_1C, 0.75B_1C]$ in section 5.3.2 [250, 500, 750, 1000, 1250, 1500] in section 5.3.3, 5.3.4 [750, 1000, 1250, 1500] in section 5.3.5, 5.3.6
c_{1ij}	$0.01\bar{l}_{ij}$	$0.01\bar{l}_{ij}$
c_2	5	5
f	1	1
$\bar{l}_k$	No limit	20
$\lvert\mathcal{S}\rvert$	1	$\lvert\mathcal{N}\rvert$
α	0.8	[0.6, 0.7, 0.8, 0.9]
Maximum computational time of subproblem	No limit	100 seconds
Maximum number of iteration	No limit	100

5.2. Algorithm performance in a small network

In this section, we conduct numerical experiments for a small network. The results show the model performance and the computational efficiency of our model compared with the SBM. The objective value and detailed computational time information are shown in Tables 5–7.

For a stochastic program, it is sufficient to generate a large number of scenarios to determine an objective value that is close to the optimum. Hence, we use different sizes of the scenario set to test the performance of the SBM. However, our row generation approach does not need scenarios initially and generates valid cuts (associated with scenarios) to obtain the optimal solutions during the solution procedure. To further improve the computational efficiency of our approach, rather than adopting the primal MP, we use the PMP model (see section 4.3) with one and five initial scenarios. Table 5 shows the average objective values, deviation between the average and minimum objective values, and deviation between the average and maximum objective values corresponding to the SBM over five different sets of scenarios. The repeated experiments show the influence of the combination of scenarios on the objective value and the computational time. These three values are shown

Table 5: The objective values in the scenario-based model and row generation algorithm

$\lvert\mathcal{OD}\rvert$	$\lvert\mathcal{T}\rvert$	SBM					RG
		Ω = 100	Ω = 200	Ω = 300	Ω = 400	Ω = 500	
20	2	(195.0000, 0.0000, 0.0000)	(195.0000, 0.0000, 0.0000)	(195.0000, 0.0000, 0.0000)	(195.0000, 0.0000, 0.0000)	(195.0000, 0.0000, 0.0000)	195.0000
40	2	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	200.0000
56	2	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	(200.0000, 0.0000, 0.0000)	200.0000
20	3	(292.9811, -0.1661, 0.2991)	(292.7508, -0.0148, 0.0395)	(292.7342, -0.0086, 0.0154)	(292.6992, -0.0782, 0.1200)	(292.6868, -0.0656, 0.1198)	292.2749
40	3	(293.1886, -0.1840, 0.4348)	(292.9921, -0.0906, 0.2638)	(292.7831, -0.1478, 0.2662)	(292.7831, -0.1478, 0.2662)	(292.7353, -0.1000, 0.2068)	292.5002
56	3	(293.1729, -0.0085, 0.0130)	(293.0244, -0.0404, 0.0733)	(292.9669, -0.0477, 0.1381)	(292.9398, -0.0252, 0.0711)	(292.9194, -0.0660, 0.1323)	292.5002
20	4	(368.4397, -2.5255, 7.4946)	(365.6615, 0.0000, 0.0000)	(365.6615, 0.0000, 0.0000)	(365.6615, 0.0000, 0.0000)	(365.6615, 0.0000, 0.0000)	365.6615
40	4	(371.2100, -2.0151, 4.8187)	(367.4155, -0.1932, 0.4795)	(367.4155, -0.1932, 0.4795)	(367.3678, -0.1455, 0.3364)	(367.3678, -0.1455, 0.3364)	367.2223
56	4	(369.3321, -1.0546, 3.0932)	(368.3993, -0.1218, 0.2949)	(367.8996, -0.2249, 0.6027)	(367.7687, -0.1963, 0.4148)	(367.7687, -0.1963, 0.4148)	367.2223

RG: Row generation approach

in brackets in Table 5. The optimal objective value of the robust model solved using a row generation algorithm 1 is shown in the last column (RG).

Although four out of a total of nine instances reach the optimal objective values by increasing the number of scenarios, the convergence of the scenario-based model cannot be ensured. When $|\mathcal{T}| = 2$, the SBM is too simple to solve. The optimal solutions are obtained even without a rebalancing operation, which is the case for $|\mathcal{OD}| = 40$ and 56. As $|\mathcal{T}|$ increases, both the number of OD demands and the dispersion degree of the bike distribution become large. Thus, the rebalancing operation is necessary to improve revenue, which increases the computational complexity of the SBM. Hence, the objective value cannot easily converge to the optimal value. We also observe that the solutions for the instances of $|\mathcal{OD}| \in \{40, 56\}$ and $\mathcal{T} \in \{3, 4\}$ are the same because the demand quantity is greater than the number of bikes allocated, and the redundant demand cannot be satisfied.

Regarding the computational time, Table 6 displays the average and the computational time interval for the scenario-based model over five repeated experiments. For each instance with a given number of scenarios, the gap between the minimum and maximum computational time is large in the five repeated experiments, which implies that the computational efficiency of solving the SBM depends highly on the composition of the scenario sets. Furthermore, the computational time increases in the number of scenarios and decision periods. Observe that when $|\mathcal{T}| = 2$, the computational time decreases as $|\mathcal{OD}|$ increases. The reason is that there is substantial demand, so the bikes already have a full load without rebalancing. As stated above, the model can be solved easily in this situation. However, when $|\mathcal{T}| > 2$, increasing $|\mathcal{OD}|$ will inevitably increase the computational time. Table 6 also lists the total computational time for the row generation algorithm. The efficiency of both of the settings of our algorithm is better than that in the scenario-based model. In addition, increasing the number of scenarios in the PMP model reduces the total computational time because increasing the number of scenarios in the PMP model will bring a requirement for a small number of iterations and the total computational time for solving the SP, which accelerates the computation (see, Table 7). This finding confirms that employing the PMP model with a suitable number of scenarios can improve the computational efficiency.

Table 6: The computational time of the SBM and row generation algorithm

$\lvert\mathcal{OD}\rvert$	$\lvert\mathcal{T}\rvert$	SBM					RG	
		Ω = 100	Ω = 200	Ω = 300	Ω = 400	Ω = 500	Ω = 1	Ω = 5
20	2	(20.73, -3.23, 2.73)	(73.15, -16.49, 13.45)	(146.42, -20.67, 31.33)	(300.57, -113.43, 104.50)	(535.09, -62.24, 100.97)	70.08	17.84
40	2	(2.62, -0.23, 0.14)	(9.59, -0.70, 0.49)	(22.04, -1.06, 1.03)	(38.64, -1.04, 1.65)	(67.88, -4.79, 3.68)	30.23	5.06
56	2	(2.52, -0.15, 0.16)	(9.26, -0.20, 0.33)	(21.84, -0.84, 1.03)	(41.65, -1.87, 1.75)	(73.37, -7.33, 7.53)	5.07	5.09
20	3	(179.67, -23.12, 14.05)	(678.43, -60.20, 56.57)	(1540.18, -150.57, 284.53)	(2925.80, -654.25, 766.17)	(4517.19, -591.88, 1058.64)	613.74	212.53
40	3	(315.29, -37.11, 21.31)	(1439.44, -41.70, 30.95)	(3404.60, -200.51, 207.66)	(6174.02, -422.61, 577.39)	(9926.88, -583.96, 696.68)	1408.97	305.77
56	3	(353.98, -14.91, 23.28)	(1287.59, -94.53, 147.65)	(3045.51, -164.30, 110.62)	(6199.13, -80.10, 66.63)	(10462.38, -1268.24, 1066.48)	1026.25	914.99
20	4	(341.38, -35.29, 50.04)	(1403.73, -350.01, 433.49)	(3082.88, -379.30, 361.89)	(5830.57, -903.89, 1091.21)	(12232.29, -832.17, 916.44)	704.61	315.43
40	4	(720.61, -96.28, 167.50)	(3590.96, -176.24, 92.15)	(8362.28, -1122.18, 1824.01)	(18145.36, -2192.18, 2656.14)	(31700.03, -2209.04, 3571.92)	878.79	317.82
56	4	(865.56, -73.61, 57.23)	(3440.20, -476.87, 480.12)	(9014.52, -2074.50, 1242.36)	(17916.89, -1504.85, 1735.73)	(27714.07, -4063.05, 4191.60)	851.63	564.36

5.3. Model performance with a real-size network

In this section, we use a real-size network to show the distribution of the station locations and the service areas. Then, we investigate the performance of a robust model associated with different parameter settings, including the number of constructed stations (B_1), the number of allocated bikes (B_2), and the maximal capacity of stations (C). The results are shown in sections 5.3.2 and 5.3.3. Furthermore, section 5.3.4 shows the performance under various conservation levels $\mathbf{\Gamma}$. Finally, we compare our deterministic and robust models with some benchmark models in sections 5.3.5 and 5.3.6.

Table 7: The number of iterations (Iter.) and the proportion of computational time for the SP (Time of SP) of the row generation algorithm

Path	T	Iter.		Time of SP (%)	
		Ω = 1	Ω = 5	Ω = 1	Ω = 5
20	2	17	4	97.87	89.35
40	2	6	1	99.57	98.81
56	2	1	1	99.01	98.23
20	3	116	36	94.66	84.82
40	3	262	48	92.50	78.64
56	3	191	127	93.28	69.54
20	4	132	50	93.82	79.39
40	4	164	44	93.29	69.41
56	4	160	74	94.19	65.76

5.3.1. The distribution of the station locations and the service areas

In this section, a realistic example is provided to show the results of the installed bike stations and the associated service area design. In this specific computational experiment, the number of stations allowed is 30, and the initial number of allocated bikes is 1000. There are some zones that are not covered by stations, and the demand in these zones is lost. In addition, we see that not all stations are covered by a service area (see the zones marked by shadow lines). This means that there is no need to perform the rebalancing operation at these stations because the inventories in these stations are sufficient to serve daily demand, and the rebalancing operation can be performed at night.

Furthermore, we observe that the stations in a service area may be dispersive (see the service areas highlighted by pink and orange). This indicates that zones with stations installed in the same service area may not be adjacent. A possible reason for this is that adjacent zones may have similar bike supply levels; i.e., these stations may all be undersupplied or oversupplied. Hence, the rebalancing operation between these stations cannot improve bike usage and satisfy a greater demand. On the other hand, if, for example, the rebalancing cost is very high compared with that of other restrictions, the ranges of all service areas will decrease, and the zones with stations installed in each service area will become closer. Hence, we recommend that when designing vehicle service areas, decision makers should consider not only the geographical distribution of stations and the values of parameters such as the rebalancing cost, maximum travel distance of vehicles and number of vehicles but also the associated supply and demand levels of bikes at these stations.

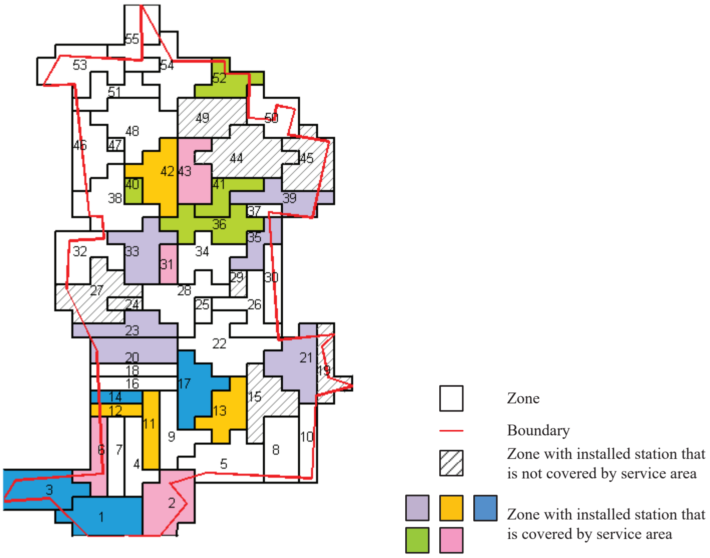
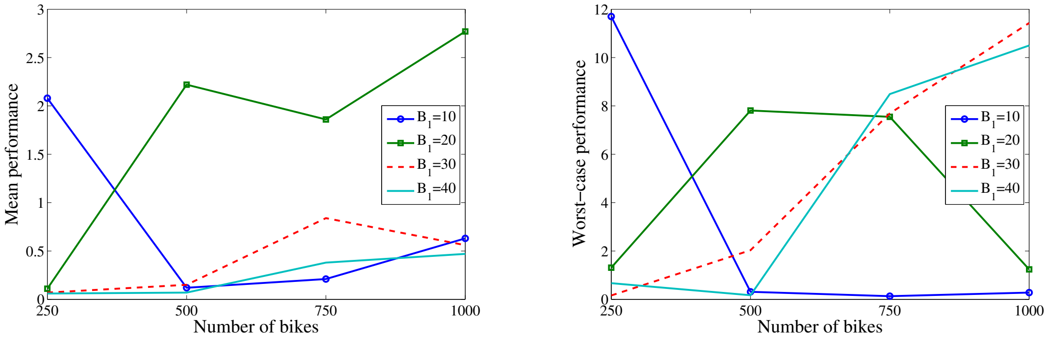

55
53
54
52
51
49
50
48
46
47
44
45
42
43
40
41
38
39
37
36
35
32
33
34
31
28
29
30
27
24
25
26
23
22
20
21
18
19
16
17
14
15
12
13
11
9
6
7
10
8
4
5
3
2
1
Zone
Boundary
Zone with installed station that is not covered by service area
Zone with installed station that is covered by service area

Figure 5: The distributions of station locations and service areas

5.3.2. The impact of B_1 and B_2

Let $B_1 \in \{10, 20, 30, 40\}$, $B_2 \in \{250, 500, 750, 1000\}$, and $C = 100$. The relationship between the numbers of stations and bikes and the performance of the robust model is shown in Fig. 6. The mean performance of the robust model solutions is relatively low. However, the worst-case performance is highly dependent on the numbers of both the station locations and the initial bikes allocated. In addition, for a given number of stations B_1, the worst-case performance trends with regard to different numbers of bikes B_2 are highly inconsistent.

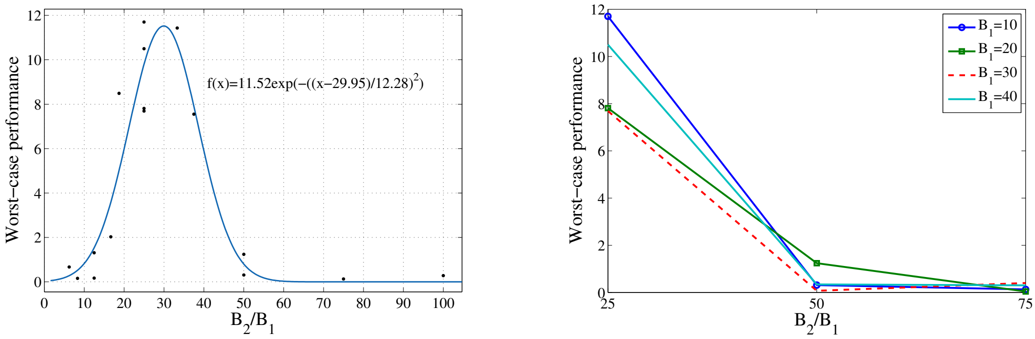


(a)

(b)

Figure 6: Performance of robust solutions with different values of B_1 and B_2 for (a) the mean performance and (b) the worst-case performance

To investigate the relationship between the worst-case performance and the numbers of stations and bikes, we define the budget ratio B_2/B_1, which represents the mean number of bikes at each station. We show the relationship between the worst-case performance and the budget ratio in Fig. 7(a). A Gaussian fitting curve is generated accordingly. The adjusted R-square is equal to 0.8891, which indicates that the curve reflects the relationship well. A practical guide to this curve is that the worst-case performance is not significant when the budget ratio is at a low or high level. When bikes are oversupplied, most of the demand can be satisfied even in the worst-case situation. Following the same logic, when bikes are undersupplied, even given low demand, corresponding to the worst-case situation, they cannot be serviced completely. Both cases show that there is little benefit to considering the robustness of the system. By comparison, the worst-case performance becomes significant when the budget ratio is at the mid-level. In this case, there are some bikes in the system but not enough, and bike firms must make decisions to assign bikes to different stations to improve the bike use and total revenue.



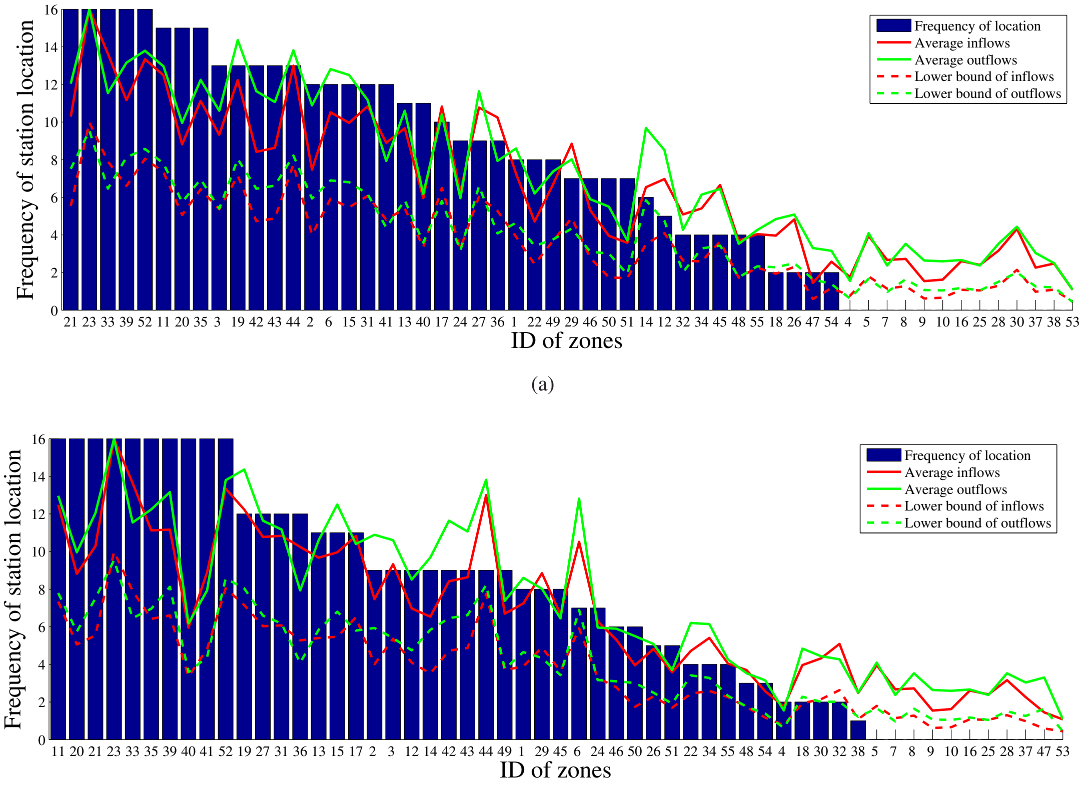
(a)

(b)

Figure 7: Worst-case performance of the robust solutions with a budget ratio

It appears that the peak worst-case performance of these instances lies in the range of 20 to 30 for the budget ratio. Further tests setting $B_2 = [0.25B_1C, 0.5B_1C, 0.75B_1C]$ also show similar results (see Fig. 7(b)). However, this peak may also depend on the detailed topological structure of the network and associated demands. We will study this peak from the perspective of theoretical and numerical analyses in future research.

Besides the model performance, we also focus on the distribution of stations in all 16 instances, both for the deterministic and robust models, as shown in Fig. 8. For convenience, we display the mean and lower bounds of the inflows and outflows when analysing the relationship between the demand and the station locations. The deterministic model selects 42 zones, five of which are always located by station in all instances. The selected zones are likely to have a substantial average demand, whereas the robust solutions locate stations in 45 zones, and 10 of them are selected in all instances. The frequencies for the station locations in some zones are quite different in the two models. For example, the stations are usually located in zones 6, 42, and 43 when employing the deterministic model but less frequently in the robust model. The reason is that more of the demand in these zones is intra-zonal. That is, the bikes in these zones can efficiently meet the local demand but reduce the negotiability of bikes, which causes lower bike utilization and decreases revenue when demand is low. Hence, we believe that the deterministic solutions focus on zones with greater demand, while the robust solution tends to assign stations to the zones with greater inter-zonal demand.



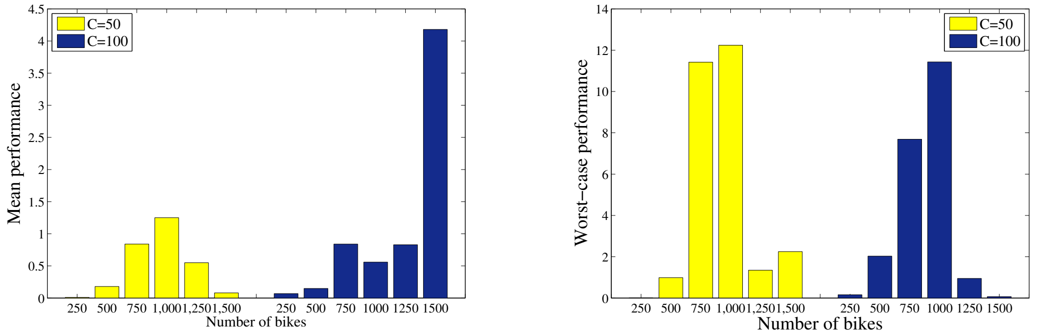
(a)

(b)

Figure 8: Distribution of station locations over all tests for (a) the deterministic model and (b) the robust model

5.3.3. The impact of the capacity parameters

The impact of the capacity on the worst-case performance is shown in Fig. 9. We conduct numerical experiments on two different capacities ($C = 50$ and $C = 100$) and different total numbers of bikes. The number of stations is 30. In general, the worst-case performance with regard to different total numbers of bikes shows a similar trend to its mean performance counterpart. The two performance indicators both increase first and then decrease except for the cases of $C = 100$ and $B_1 = 1500$. The best worst-case performance corresponds to a total number of bikes of approximately 750 or 1,000. This result indicates that different capacity levels have similar impacts on the worst-case performance. In addition, these results show that the robust model works well when the number of bikes is at a middle level.



(a)

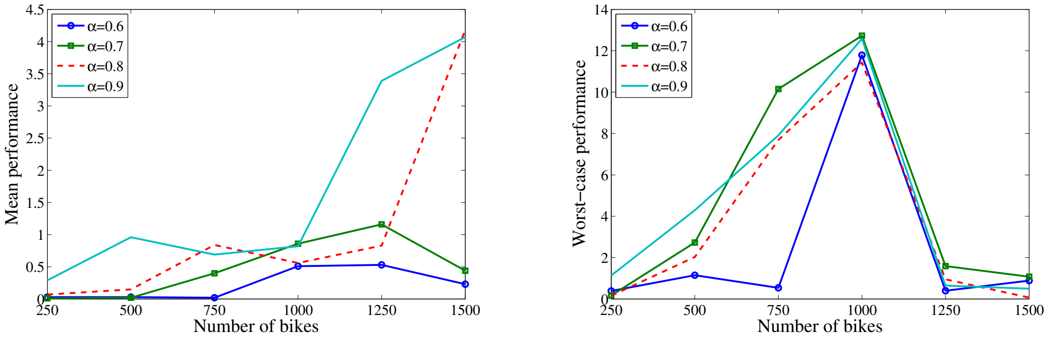
(b)

Figure 9: Performance associated with various station capacities for (a) the mean performance and (b) the worst-case performance

5.3.4. *The impact of conservation levels*

After analyzing the impact of some parameters (B_1, B_2 and C) on model performance, in this section, we discuss the impact of conservation levels. The value of $\mathbf{\Gamma}$ represents the conservation level of the bike-sharing firms under uncertainty. The larger the value of α is, the more conservative the decision makers are. The relationship between α and $\mathbf{\Gamma}$ is referred to as the *setting of the uncertainty set* in section 5.1.1. We set the parameters $B_1 = 30$, $B_2 \in \{250, 500, 750, 1000, 1250, 1500\}$, and $C = 100$, and the value of α ranges from 0.6 to 0.9 with an interval step of 0.1. Then, the associated mean and worst-case performance are shown in Fig. 10.

The influence of different conservation levels on the worst-case performance exhibits a similar trend to that of the total number of bikes. In general, the value of α does not cause a significant difference in the curve trend in Fig. 10(b). However, a large α worsens the mean performance when the total number of bikes is large. The managerial insights supported by these results are that firms should not be too pessimistic about demand uncertainty, which causes a low-yield strategic scheme without a significant improvement in robustness. Hence, decision makers should not consider an overly conservative demand level to ensure a high average revenue. Relative to the budget ratio (B_2/B_1), which has a considerable influence on the tendency of the worst-case performance, the conservation level has a greater impact on the magnitude of the worst-case performance.



(a)

(b)

Figure 10: Performance of robust solutions with different conservation levels for (a) the mean performance and (b) the worst-case performance

5.3.5. *Integration vs. separation*

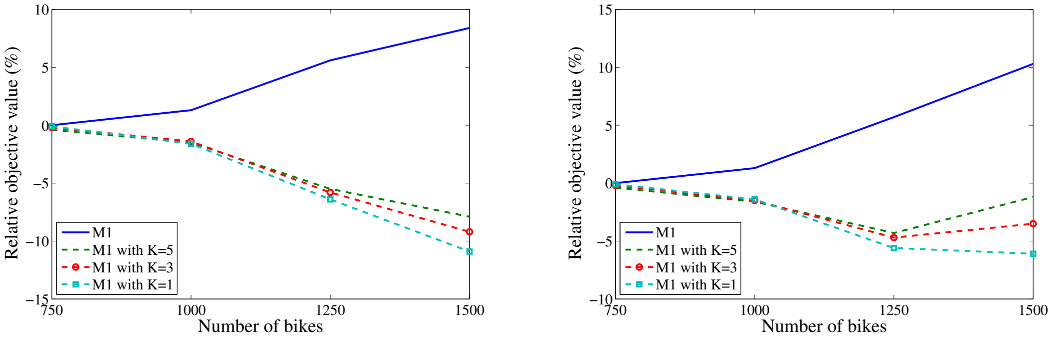
In this section, we analyze the necessity of jointly optimizing the station location and service area design. Hence, we compare our integrated deterministic model with the benchmark model that determines the station location and service area design separately.

In detail, we consider two benchmark models that are related to two sequential decision phases. In the first phase, we optimize the station location without the relocation restriction by the service areas. This model is named M1. M1 is similar to our deterministic model without constraints (4)–(14) (see the following model). It means that the rebalance operation can be conducted between any two stations.

$$\begin{aligned} \max \ & f\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} w_{ijt} - \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{t\in\mathcal{T}} c_{1ij}r_{ijt} \\ \text{s.t.} \ & (1)-(3),(15)-(17) \\ & r_{ijt} \leq Q, \forall i,j \in \mathcal{N},\ t\in\mathcal{T}, \\ & \boldsymbol{x} \in \{0,1\}, \boldsymbol{v},\boldsymbol{r},\boldsymbol{w} \geq 0. \end{aligned}$$

Clearly, the objective value of the M1 is larger than that of our deterministic model because of the fewer constraints imposed. Fig. 11 shows that M1 overestimates the true revenue of the system, especially when the number of bikes is large (e.g., $B_2 = 1500$).

In the second phase, we employ a maximal covering facility location model (namely, M2(K)) to determine K service areas of the stations chosen by model M1. This model is reasonable since the stations with large demand could have high priority to be covered by rebalancing vehicles. The maximal travel distance in a service area is also restricted by a threshold $\bar{l}_k$. We re-estimate the revenue of the solutions solved by M1 with the distributions of service areas obtained from M2 (see the dotted lines in Fig. 11). The results show that using the service areas designed by M2 will restrict the fleet rebalancing operation and reduce the objective value of M1 to less than that of our model when the number of bikes is large. Moreover, the results of M2 demonstrate that optimizing the number and distribution of service areas can be more important than providing more bikes.



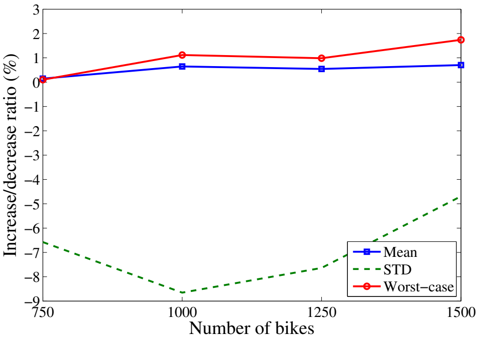
(a)

(b)

Figure 11: Relative revenue between benchmark approach and our deterministic model under (a) $C = 50$ and (b) $C = 100$

5.3.6. Comparison between robust optimization and stochastic program

From the perspective of the methodology, the stochastic program and robust optimization approach are both powerful modeling tools in the context of uncertainty. In this section, we compare the performance between our robust model and its stochastic program counterpart. The two-stage stochastic program model aims to maximize the expected revenue based on our seven-day demand data (i.e., we use seven scenarios in the stochastic model). These real data are also used to construct the uncertainty set for robust optimization. The conservation level α of our robust model is set to 0.7. The out-of-sample revenues of the robust and stochastic models are generated by the simulation process in section 5.1.1. Fig. 12 exhibits the relative ratios of the mean, STD and worst-case between the out-of-sample revenues of the robust model and the stochastic model. It implies that the robust model can ensure higher mean and worst-case revenue than the stochastic model with greater revenue stability. That is, compared with that of the stochastic model, the robust model can slightly increase the mean revenue by 0.14%-0.7%, improve the worst-case revenue by 0.1%-1.7%, and significantly decrease the STD by 4.6%-8.6%. The reason is that the stochastic model constructed with a small number of scenarios will overfit the in-sample revenue, and the solutions can have both low robustness and low out-of-sample revenue. Hence, the robust optimization approach is reasonable to optimize the system effectiveness under the condition that sufficient reliable data are not available.



(a)

Figure 12: Relative ratios of the mean, STD and worst-case between the out-of-sample revenues of the robust model and the stochastic model

6. Conclusion

In our study, we investigate a station location problem for bike-sharing systems integrated with rebalancing vehicle scheduling. Our goal is to maximize the daily revenue minus the rebalancing cost under a given number of stations and bikes. Existing studies on station location problems typically consider the quantity of the rebalancing operation with the flow conservation constraint or do not consider it at all. Our work determines the positions of stations, their initial inventories, and the service areas corresponding to the rebalancing vehicles. Each service area with a limited maximum radius is composed of several stations. We assume that adopting service areas to depict the rebalancing operation satisfies the requirements of the firms' management.

Because of the widespread phenomena of data loss, small data sizes, and data errors, the ambiguity of demand impedes the optimal decision process. Hence, we define an uncertainty set to describe uncertain demand and employ a two-stage robust framework to solve this problem. However, the associated deterministic equivalent formulation is an intractable semi-infinite model. To solve this model efficiently, the primal model is first separated into two parts: the master problem and the subproblem. Then, a customized row generation approach is used to solve these two problems iteratively and obtain the optimal solutions to the original model.

We design extensive numerical experiments to verify the efficiency of our model and algorithm. We use the instances in the small network to perform our row-generation algorithm. Compared with a scenario-based model, our algorithm obtains better solutions in less computational time. In addition, we design several instances based on a large real-size network to test the performance of the robust solutions. In contrast to the solutions of the deterministic model, the associated results of the robust solutions reveal that there exists an approximate 10% worst-case performance improvement with a small cost in terms of the mean performance when the budget ratio (i.e., the average number of bikes per station) is mid-level. However, if the systems have a much smaller or larger budget ratio, the robustness of the strategic scheme using the robust model is no different than that using the deterministic model. In addition, a dispersive distribution of stations in each service area implies that the optimal fleet rebalancing operation does not have to be confined to one geographical area. Furthermore, our robust model exhibits better performance (e.g., larger mean and worst-case revenues and a higher revenue stability) than the stochastic model with a small data set.

There are several further studies that can be conducted in the future. (i) Although the solutions to the large-scale problem are satisfactory, they are not optimal. A more effective algorithm to handle large-scale network problems is needed. (ii) A more general description of the uncertainty set is required, such as the mean-variance information. (iii) Concerning the methodology, two-stage robust optimization can be used in other transportation problems. (iv) Problem-wise, a more general model that there are multiple stations in one zone could be considered. Therefore, one more level of optimization should be added between stations and zones for both flow assignment and rebalancing operations. Related computation problem should be solved as well.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under projects 71971154, 72010107004, 71890972 and 71890970, the MOE (Ministry of Education in China) Project of Humanities and Social Sciences (Project No.17YJC630239), and the UK Department for Transport ("Future Street" project).

Appendix A. Zone generation procedure

In appendix A, we show the zone generation procedure. The result of this procedure is the input OD pair information for our model. Note that the notation systems in appendix A are independent of those in the main body of our paper.

The actual bike-sharing system data reveal highly dispersive distributions of demand *points* and OD pairs. Using many *grids* to cover all demand is a common practice to generate OD pairs. In our study, we rasterize the enclosing rectangle of Dongcheng District, Beijing, China, into 40 × 20 rectangular grids, with 483 such grids covering the entire *region*. Thus, there are over 10,000 OD pairs over seven days in the investigated *region*. The maximum flow of all the OD pairs is less than 50. This feature is derived from the free-floating mobility of Mobike. Using these data as the input for our model immediately results in consuming substantial computational time. To overcome this difficulty, a mathematical programming model is proposed to merge the grids into a larger *zone* so that fewer OD pairs are generated. In the meantime, this procedure aims to reduce the loss of OD information. The merged grids are called a *zone*. Customers in each zone are assumed to access the station in the zone. The size relationship among these concepts is "point < grid < zone < region". The principles of the zone generation procedure are discussed below before providing the mathematical programming model.

(i) Each zone has a size limit. To ensure that customers can only access systems in their own zone, the maximum radius of each zone should be less than the threshold. Otherwise, the cost of searching for available bikes in a zone could be high.

(ii) Each zone should ensure its connectivity. For a specific zone, all the grids in the zone should be connected.

(iii) The OD information loss within all zones should be minimal. By merging the grids in the zone, it is inevitable that some OD information will be lost within each zone. Therefore, a basic strategy is to restrict these losses.

There are many approaches that can be used to conduct the merge operation. In our study, we propose an optimization-based approach. The mathematical programming model is created in terms of the abovementioned principles.

(i) For each zone, we design a concept of the "centre". The centre corresponds to a grid in the zone. The distance between each grid in this zone and the centre should be less than or equal to the threshold. The distance is defined as $\bar{l}_{ik}$ between grids i and k. In our study, the Manhattan distance is adopted. Note that other distance measures can also be used.

(ii) Connectivity constraints should be mathematically guaranteed. A concept of "neighbour" is introduced. The neighbour grid set of grid i is defined as the grids that have a common edge with grid i. Hence, a rectangular grid has at most four neighbours. If grid k is the centre grid of a zone and grid i is also in this zone, there must exist a neighbour of i, denoted as j, also in this zone. Otherwise, grid i is not connected to grid k, so the zone itself is not connected. Hence, grid j is named the *upstream grid* of grid i if grid j satisfies the criteria that (a) it is in the neighbour set of i, (b) $\bar{l}_{jk} = \bar{l}_{ik} - 1$, and (c) grids i and j are in the same zone.

(iii) It is relatively easy to implement the third principle by limiting the total demand in the same origins and destinations under a given threshold. The idea of the mathematical formulations is omitted here.

We can now formulate the mixed integer programming model. Let $\mathcal{N}'$ be a set of grids and $\mathcal{S}_i$ be the neighbour set of grid i. D_{ij} is defined as the travel demand from grid i to j. We define the binary variable x_{ik} as 1 if grid i is in the same zone as grid k, where k is the centre of the zone, and 0 otherwise. Thus, $x_{ii} = 1$ if i is a centre. There are two groups of additional decision variables, y_{ijk} and z_{ijk}. y_{ijk} equals 1 if grid j is the upstream grid of i in the same zone as centre k and 0 otherwise. The decision variable z_{ijk} is used to state whether grids i and j are in the same zone as the centre of grid k. Our objective is to minimize the number of zones (i.e., the sum of x_{ii}) without losing too much trip information. The formulation is as follows:

$$\min \sum_{i\in\mathcal{N}'} x_{ii} \tag{60}$$

$$s.t. \sum_{k\in\mathcal{N}'} x_{ik} = 1, \qquad \forall i \in \mathcal{N}', \tag{61}$$

$$x_{ik} \leq x_{kk}, \qquad \forall i,k \in \mathcal{N}', \tag{62}$$

$$z_{ijk} \leq x_{ik}, \qquad \forall i,j,k \in \mathcal{N}', \tag{63}$$

$$z_{ijk} \leq x_{jk}, \qquad \forall i,j,k \in \mathcal{N}', \tag{64}$$

$$z_{ijk} \geq x_{ik} + x_{jk} - 1, \qquad \forall i,j,k \in \mathcal{N}', \tag{65}$$

$$\sum_{k\in\mathcal{N}'}\sum_{i\in\mathcal{N}'}\sum_{j\in\mathcal{N}'} D_{ij}c_{ijk} \leq \gamma' \sum_{i\in\mathcal{N}'}\sum_{j\in\mathcal{N}'} D_{ij}, \tag{66}$$

$$M(1 - y_{ijk}) \geq \bar{l}_{jk} - \bar{l}_{ik} + 1, \qquad \forall i,k \in \mathcal{N}', i \neq k, j \in \mathcal{S}_i, \tag{67}$$

$$y_{ijk} \leq x_{jk}, \forall i,k \in \mathcal{N}', i \neq k, \qquad \forall j \in \mathcal{S}_i, \tag{68}$$

$$\sum_{j\in\mathcal{S}_i} y_{ijk} = x_{ik}, \qquad \forall i,k \in \mathcal{N}', i \neq k, \tag{69}$$

$$\bar{l}_{ij} z_{ijk} \leq \bar{l}_k, \qquad \forall i,j,k \in \mathcal{N}', \tag{70}$$

$$x_{ik}, y_{ijk}, z_{ijk} \in \{0,1\}, \qquad \forall i,j,k \in \mathcal{N}'. \tag{71}$$

Constraint (61) restricts each grid so that it can only belong to a single zone, and constraint (62) limits the relationship between grid i and centre k. Constraints (63)–(65) define whether grids i and j are in the same zone as centre k ($z_{ijk} = 1$) or not ($z_{ijk} = 0$). Constraint (66) ensures that the total OD flow loss does not exceed the threshold. Constraints (67)–(69) ensure the connectivity of each zone.

For a given grid i in a zone with centre $k, k \neq i$, constraints (68) and (69) ensure that there exists a neighbour grid j of i and that that grid is in the same zone as i. Constraint (67) states that when the value of the right-hand side is 0, grid j is the upstream grid of grid i. Hence, all grids, except the centre k of the zone, must contain an upstream grid so that the zone is connected. Constraint (70) restricts the size of the area of each zone. This constraint accords with the fact that the service radii of bike-sharing stations cannot be large. Constraint (71) is a binary constraint.

In practice, solving the large-scale problem using a solver is time-consuming. We first separate the region into 10 small-scale subregions, each of which includes approximately 50 grids, and then solve them one by one. The result is composed of 55 zones in the region based on the parameters $\gamma' = 0.15$ and $\bar{l}_k = 5$. The zone distribution is shown in Fig. 13.

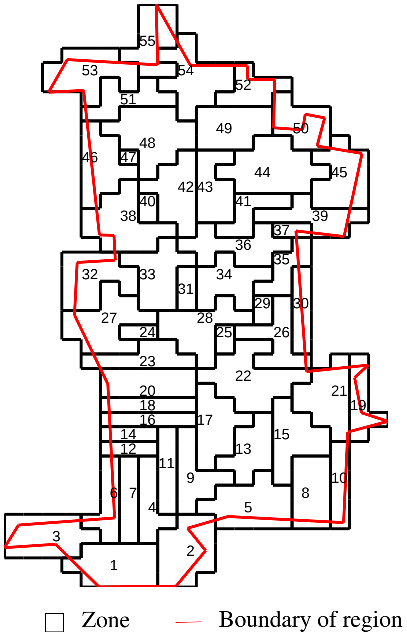


Figure 13: Zone distribution in Dongcheng District

References

Albareda-Sambola, M., Fernández, E., Laporte, G., 2007. Heuristic and lower bound for a stochastic location-routing problem. European Journal of Operational Research 179 (3), 940–955.

Albareda-Sambola, M., Fernández, E., Nickel, S., 2012. Multiperiod location-routing with decoupled time scales. European Journal of Operational Research 217 (2), 248–258.

Basciftci, B., Ahmed, S., Shen, S., 2019. Distributionally robust facility location problem under decision-dependent stochastic demand. arXiv preprint arXiv:1912.05577.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., Rennen, G., 2013. Robust solutions of optimization problems affected by uncertain probabilities. Management Science 59 (2), 341–357.

Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust optimization. Princeton University Press.

Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A., 2004. Adjustable robust solutions of uncertain linear programs. Mathematical programming 99 (2), 351–376.

Ben-Tal, A., Nemirovski, A., 1997. Robust truss topology design via semidefinite programming. SIAM Journal on Optimization 7(4), 991–1016.

Ben-Tal, A., Nemirovski, A., 1998. Robust convex optimization. Mathematics of operations research 23(4), 769–805.

Bertsimas, D., Brown, D. B., Caramanis, C., 2011. Theory and applications of robust optimization. SIAM review 53 (3), 464–501.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operations Research 52 (1), 35–53.

Bertsimas, D., Sim, M., Zhang, M., 2018. Adaptive distributionally robust optimization. Management Science.

Birge, J. R., Louveaux, F., 2011. Introduction to stochastic programming. Springer Science & Business Media.

Boyacı, B., Zografos, K. G., Geroliminis, N., 2015. An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research 240 (3), 718–733.

Brandstätter, G., Kahr, M., Leitner, M., 2017. Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand. Transportation Research Part B: Methodological 104, 17–35.

Bulhões, T., Subramanian, A., Erdoğan, G., Laporte, G., 2018. The static bike relocation problem with multiple vehicles and visits. European Journal of Operational Research 264, 508–523.

Çelebi, D., Yörüsün, A., Işık, H., 2018. Bicycle sharing system design with capacity allocations. Transportation research part B: methodological 114, 86–98.

Chan, T. C. Y., Shen, Z. J. M., Siddiq, A., 2018. Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation. Operations Research 66 (2), 358–379.

Chang, J., Yu, M., Shen, S., Xu, M., 2017. Location design and relocation of a mixed car-sharing fleet with a co2 emission constraint. Service Science 9 (3), 205–218.

Contardo, C., Rousseau, L. M., Morency, C., 2012. Balancing a dynamic public bike-sharing system. Montreal: Cirrelt 4.

Crainic, T. G., Hewitt, M., Maggioni, F., Walter, R., 2016. Partial benders decomposition strategies for two-stage stochastic integer programs. https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2016-37.pdf.

Delage, E., Ye, Y., 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations research 58 (3), 595–612.

Dell'Amico, M., Iori, M., Novellani, S., Subramanian, A., 2018. The bike sharing rebalancing problem with stochastic demands. Transportation research part B: methodological 118, 362–380.

Demaio, P., 2009. Bike-sharing: History, impacts, models of provision, and future. Journal of Public Transportation 12 (4).

Drexl, M., Schneider, M., 2015. A survey of variants and extensions of the location-routing problem. European Journal of Operational Research 241 (2), 283–308.

Dunlap, R., Jorgenson, A., 2012. Environmental problems. The Wiley-Blackwell Encyclopedia of Globalization.

Erdoğan, G., Laporte, G., Calvo, R. W., 2014. The static bicycle relocation problem with demand intervals. European Journal of Operational Research 238 (2), 451–457.

Forma, I. A., Raviv, T., Tzur, M., 2015. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. Transportation Research Part B: Methodological 71, 230–247.

Frade, I., Ribeiro, A., 2015. Bike-sharing stations: A maximal covering location approach. Transportation Research Part A: Policy and Practice 82, 216–227.

Francis, P. M., Smilowitz, K. R., Tzur, M., 2008. The period vehicle routing problem and its extensions. In: The vehicle routing problem: latest advances and new challenges. Springer, pp. 73–102.

Garcìa-Palomares, J. C., Gutiérrez, J., Latorre, M., 2012. Optimizing the location of stations in bike-sharing programs: A gis approach. Applied Geography 35 (1C2), 235–246.

Goh, J., Sim, M., 2010. Distributionally robust optimization and its tractable approximations. Operations research 58 (4-part-1), 902–917.

Gülpınar, N., Pachamanova, D., Çanakoğlu, E., 2013. Robust strategies for facility location under uncertainty. European Journal of Operational Research 225 (1), 21–35.

Haider, Z., Nikolaev, A., Kang, J. E., Kwon, C., 2018. Inventory rebalancing through pricing in public bike sharing systems. European Journal of Operational Research 270 (1), 103–117.

Hao, Z., He, L., Hu, Z., Jiang, J., 2020. Robust vehicle pre-allocation with uncertain covariates. Production and Operations Management 29 (4), 955–972.

He, L., Hu, Z., Zhang, M., 2020. Robust repositioning for vehicle sharing. Manufacturing & Service Operations Management 22 (2), 241–256.

He, L., Mak, H. Y., Rong, Y., Shen, Z. J. M., 2017. Service region design for urban electric vehicle sharing systems. Social Science Electronic Publishing 19 (2), 309–327.

Ho, S. C., Szeto, W. Y., 2017. A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem. Transportation Research Part B: Methodological 95, 340–363.

Jaillet, P., Qi, J., Sim, M., 2016. Routing optimization under uncertainty. Operations research 64 (1), 186–200.

Jensen, P., Rouquier, J. B., Ovtracht, N., Robardet, C., 2010. Characterizing the speed and paths of shared bicycle use in lyon. Transportation Research Part D: Transport and Environment 15 (8), 522–524.

Kabra, A., Belavina, E., Girotra, K., 2019. Bike-share systems: Accessibility and availability. Management Science.

Karaoglan, I., Altiparmak, F., Kara, I., Dengiz, B., 2012. The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. Omega 40 (4), 465–477.

Kaspi, M., Raviv, T., Tzur, M., 2014. Parking reservation policies in one-way vehicle sharing systems. Transportation Research Part B: Methodological 62, 35–50.

Kloimüllner, C., Papazek, P., Hu, B., Raidl, G. R., 2014. Balancing bicycle sharing systems: An approach for the dynamic case. In: European Conference on Evolutionary Computation in Combinatorial Optimization. pp. 73–84.

Laporte, G., Meunier, F., Calvo, R. W., 2015. Shared mobility systems. 4OR 13 (4), 341–360.

Legros, B., 2019. Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. European Journal of Operational Research 272 (2), 740–753.

Lei, C., Ouyang, Y., 2018. Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems. Transportation Research Part B: Methodological 109, 90–109.

Li, H., Wang, Q., Wang, H., Deng, Z., Shi, W., 2015. Residential clustering and spatial access to public services in shanghai. Habitat International 46, 119–129.

Li, Y., Szeto, W. Y., Long, J., Shui, C. S., 2016. A multiple type bike repositioning problem. Transportation Research Part B: Methodological 90, 263–278.

Lin, J. R., Yang, T. H., 2011. Strategic design of public bicycle sharing systems with service level constraints. Transportation Research Part E: Logistics and Transportation Review 47 (2), 284–294.

Lin, J. R., Yang, T. H., Chang, Y. C., 2013. A hub location inventory model for bicycle sharing system design: Formulation and solution. Computers & Industrial Engineering 65 (1), 77–86.

Liu, S., He, L., Shen, Z.-J. M., 2018. Data-driven order assignment for last mile delivery. Available at SSRN 3179994.

Liu, Z., Song, Z., 2017. Robust planning of dynamic wireless charging infrastructure for battery electric buses. Transportation Research Part C: Emerging Technologies 83, 77–103.

Lu, M., Chen, Z., Shen, S., 2017. Optimizing the profitability and quality of service in carshare systems under demand uncertainty. Manufacturing & Service Operations Management, https://doi.org/10.1287/msom.2017.0644.

Maggioni, F., Cagnolari, M., Bertazzi, L., Wallace, S. W., 2019. Stochastic optimization models for a bike-sharing problem with transshipment. European Journal of Operational Research 276 (1), 272–283.

Nair, R., Miller-Hooks, E., 2011. Fleet management for vehicle sharing operations. Transportation Science 45 (4), 524–540.

Nair, R., Miller-Hooks, E., 2014. Equilibrium network design of shared-vehicle systems. European Journal of Operational Research 235 (1), 47–61.

Pal, A., Zhang, Y., 2017. Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. Transportation Research Part C: Emerging Technologies 80, 92–116.

Park, C., Sohn, S. Y., 2017. An optimization approach for the placement of bicycle-sharing stations to reduce short car trips: An application to the city of seoul. Transportation Research Part A: Policy and Practice 105, 154–166.

Pfrommer, J., Warrington, J., Schildbach, G., Morari, M., 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Transactions on Intelligent Transportation Systems 15 (4), 1567–1578.

Qi, J., Sim, M., Sun, D., Yuan, X., 2016. Preferences for travel time under risk and ambiguity: Implications in path selection and network equilibrium. Transportation Research Part B: Methodological 94, 264–284.

Saif, A., Delage, E., 2020. Data-driven distributionally robust capacitated facility location problem. European Journal of Operational Research.

Schuijbroek, J., Hampshire, R. C., Hoeve, W. J. V., 2017. Inventory rebalancing and vehicle routing in bike sharing systems. European Journal of Operational Research 257 (3), 992–1004.

Shaheen, S. A., Guzman, S., Zhang, H., 2010. Bikesharing in europe, the americas, and asia: Past, present, and future. Transportation Research Record: Journal of the Transportation Research Board 2143 (1316350), 159–167.

Shu, Jia, Chou, Mabel, C., Liu, Q., Teo, C., Wang, I., 2013. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. Operations Research 61 (6), 1346–1359.

Shui, C. S., Szeto, W. Y., 2017. Dynamic green bike repositioning problem a hybrid rolling horizon artificial bee colony algorithm approach. Transportation Research Part D: Transport and Environment, http://dx.doi.org/10.1016/j.trd.2017.06.023.

Szeto, W. Y., Liu, Y., Ho, S. C., 2016. Chemical reaction optimization for solving a static bike repositioning problem. Transportation Research Part D: Transport and Environment 47, 104–135.

Tal, R., Ofer, K., 2013. Optimal inventory management of a bike-sharing station. IIE Transactions 45 (10), 1077–1093.

Wang, J., Lindsey, G., 2019. Do new bike share stations increase member use: A quasi-experimental study. Transportation Research Part A:Policy and Practice, 1–11.

Warrington, J., Ruchti, D., 2019. Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems. Transportation Research Part C: Emerging Technologies 104, 110–134.

Weikl, S., Bogenberger, K., 2013. Relocation strategies and algorithms for free-floating car sharing systems. IEEE Intelligent Transportation Systems Magazine 5 (4), 100–111.

Wiesemann, W., Kuhn, D., Sim, M., 2014. Distributionally robust convex optimization. Operations Research 62 (6), 1358–1376.

Zeng, B., Zhao, L., 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. Operations Research Letters 41 (5), 457–461.

Zhang, D., Yu, C., Desai, J., Lau, H. Y. K., Srivathsan, S., 2017. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. Transportation Research Part B: Methodological 103, 188–207.

Zhang, Y., Baldacci, R., Sim, M., Tang, J., 2019. Routing optimization with time windows under uncertainty. Mathematical Programming 175 (1-2), 263–305.