*Rebalancing stochastic demands for bike-sharing networks with multi-scenario characteristics*

Ma, Guanhua ; Zhang, Bowen ; Shang, Changjing; Shen, Qiang

*Published in:*
Information Sciences

*DOI:*
10.1016/j.ins.2020.12.044

*Publication date:*
2021

*Citation for published version (APA):*
Ma, G., Zhang, B., Shang, C., & Shen, Q. (2021). Rebalancing stochastic demands for bike-sharing networks with multi-scenario characteristics. *Information Sciences*, *554*, 177-197. https://doi.org/10.1016/j.ins.2020.12.044

# Rebalancing Stochastic Demands for Bike-sharing Networks with Multi-scenario Characteristics

Guanhua Ma[a], Bowen Zhang[b,*], Changjing Shang[c], Qiang Shen[c]

[a] *School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China*
[b] *School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China*
[c] *Department of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, U.K.*

## Abstract

Bike-sharing networks have become a carbon-emission and environmentally friendly form of transportation in recent years. However, the asymmetric demand patterns of user behaviour, both temporally and spatially, inevitably lead to an imbalance in the distribution of shared bikes in cities, thereby becoming the greatest obstacle to the networks' development. Based on the real-world data of cycling trips, we analyse the challenging problem of imbalanced bike distribution from the entire-city perspective, establishing that the static rebalancing demand for the whole city is a stochastic variable with multi-scenario characteristics. On this basis, we develop an integer programming model to consider multiple rebalancing vehicles with time-varying rental costs, to alleviate the imbalanced bike distribution, while also analysing the intrinsic properties of such a model. We further propose a chance constraint programming model, optimising a bike-sharing network through the implementation of various genetic algorithms that employ block crossover and variable mutation operators. We reveal the inability of deterministic models in addressing the real-world problem of rebalancing demands for operational bike-sharing. In the meantime, supported with stochastic simulation, we demonstrate that the proposed approach can resolve this problem both effectively and efficiently, ensuring the delivery of a high-level bike-sharing service across an entire metropolitan city.

*Keywords:* *Bike sharing network, Static rebalancing operations, Integer programming model, Genetic algorithms*

## 1. Introduction

Bike-sharing networks first emerged as an alternative mode of public transport in the 1960s. They have received worldwide attention in the past decade due to the widespread use of smart devices and the rapid development of information technology, such as smart phones, Internet of Things (IoT) and mobile Internet. Today, bike-sharing networks are in operation throughout the world, offering health benefits together with a zero-emission and environmentally friendly mode of transport. Bike sharing is particularly convenient for the beginning and end segments of commuters'

[*] Corresponding author
*Email address:* zhang-bowen@foxmail.com (Bowen Zhang)

daily journeys. Both young people and high-income elites have increasingly adapted to transformation from internal-combustion-powered cars to shared bikes [1], and many commuters who used to take buses for short travel have switched to shared bikes [2]. Indeed, bike-sharing networks have made significant contributions to reducing urban carbon emissions and the positive transformation of commuters' daily travel habits [3].

Depending on the underlying form of service, bike-sharing networks can be divided into station-based and free-floating ones. In a station-based network, the start and end points of each cycling trip are limited to pre-selected parking places, which is conducive to centralised operation and management from the perspective of the company but implies the difficult task of planning the location and capacity of parking places. Such networks include Vélib' in France, Citi Bike in the United States and Hangzhou Public Bicycle in China. A free-floating network, without suffering from the aforementioned problem, allows users to start and finish their trips almost anywhere in the city, which greatly improves user satisfaction but necessitates the hugely costly operation of rebalancing the shared bikes, that is, transporting them to where they are needed using vehicles. Such networks include YoBike in England, oBike in Singapore and Mobike in China. Although both forms of bike-sharing face their own dilemmas, they inevitably encounter one common challenge in the asymmetric demand pattern of users across the city where a bike-sharing network is set up [4].

This asymmetric demand is mainly manifested in two aspects. First, with the continuous expansion of urban boundaries and the increasingly clear division between residential, commercial and industrial areas, citizens' home-to-work distances have become increasingly long, resulting in significant differences between the demand patterns in the different areas and times of service. For example, in the morning, the cycling demand is overwhelmingly from homes to bus stops or metro/light rail stations, and in the evening the trend is exactly the opposite. Second, from the perspective of the entire city, the demand patterns of citizens in a given area change over time due to the stochasticity and volatility of citizens' daily activities [5]. For example, instead of taking the bus or metro/light rail after cycling as usual, certain commuters may choose to take a taxi directly to work due to their late departure. Therefore, asymmetric demand patterns, both spatially and temporally, are inevitable in a bike-sharing network. Without the help of rebalancing operations, asymmetric demand patterns will unavoidably lead to an imbalance in the distribution of shared bikes in the network across the serviced city.

Imbalanced bike distribution is reflected by the accumulation of bikes in some parking places and the shortage of bikes in others, which is highly detrimental to the sustainable operation of a bike-sharing network for various reasons. For instance, if a commuter cannot find an available shared bike at the beginning of their intended cycling trip, or a vacant parking place at the end the trip, this commuter's satisfaction with the network service will be severely affected adversely. Also, the massive accumulation of shared bikes will undoubtedly worsen the existing problem of limited public parking space and cause problems for the urban environment. What is worse, particularly for free-floating bike-sharing networks, the location distribution of bikes is also influenced by users' disordered parking behaviours, making the challenge of imbalanced distribution even more difficult to deal with [6]. These problems hinder the development and application of bike-sharing networks, deterring their wider acceptance by commuters and their further integration into traditional public transport systems.

With the development and application of big data technology in data analysis and decision optimization [7, 8], extensive studies have been conducted in the field of operations and management science to enhance the ability of bike-sharing networks to tackle the problem of imbalanced distribution. These studies may be categorised into three hierarchical levels, including strategic design, tactical development and operations management [9]. In terms of strategic design, research has been aimed to ensure satisfactory levels of service and inventory in bike-sharing networks through rational planning of the location and capacity of the stations and/or parking places, thereby reducing user dissatisfaction, station construction cost, or both, e.g., [10, 11, 12, 13, 14, 15]. The studies at the tactical level have focused on formulating various user incentive policies in an effort to encourage users to ride shared bikes from oversupplied areas to undersupplied areas, e.g., [16, 17, 18]. Both tactical and strategic-level studies are conducted before the bike-sharing network under investigation begins to operate. This makes it difficult to plan for appropriate response to any imbalance between supply and demand resulting from short-term changes in shared bike use.

In contrast, research at the operations level usually involves the use of rebalancing vehicles to rectify identified imbalanced bike distribution after the network is in service. In such cases, companies use rebalancing vehicles within their operational area to pick up bikes from the oversupplied parking places and deliver them to the undersupplied parking places. In the following, for simplicity, we refer to such an operation as rebalancing. Enjoying the inherent advantages of flexibility, adaptability and universality, rebalancing has become the most popular approach to imbalanced bike distribution in recent years. The rebalancing operations described in the literature can be classified into two groups: dynamic and static. Dynamic rebalancing operations are mostly implemented during the daytime, when the distribution of the shared bikes depends dynamically on the users' cycling demand [19, 20, 21, 22]. Static rebalancing is typically performed late at night, when the shared bike usage is almost zero, so the effect of user demand is negligible. Static rebalancing operations have been much more extensively studied than dynamic ones, and they can be considered a special case of dynamic operations in which the rebalancing time period is long and the influence of bike usage is negligible. In this sense, research into static operations is the foundation of research into dynamic operations. Moreover, because static operations occur after the evening peak time and before the morning rush hours, they provide the operational team of the service company with sufficient time to effectively alleviate the two most prominent imbalances in all-day operations. For these reasons, in this paper, we mainly focus on the static rebalancing operations of bike-sharing networks.

Existing studies on static rebalancing operations can be summarised as follows. Raviv et al. [23] first introduced a static rebalancing model, and on this basis, they proposed two mixed-integer linear programming models to minimise the penalty and operating costs of a network. Ho and Szeto [24] examined rebalancing operations by reducing the total unsatisfied demand of all parking places and presented an iterative tabu search heuristic to solve the rebalancing problem. To reduce the problem scale, Forma et al. [25] resolved the rebalancing operations with a three-step heuristic, via first clustering the bike stations according to their geographic and inventory status. Szeto et al. [26] investigated the single-vehicle static rebalancing problem with the objective of minimising the weighted sum of unmet customer demand and operational time of the rebalancing vehicles. Cruz et al. [27] also addressed the rebalancing problem with a single vehicle, which allows it to carry out multiple visits to the same station while regarding each station as a temporary warehouse to store shared bikes. On this basis, they further proposed an iterative local search heuristic algorithm to resolve their rebalancing model. Li et al. [28] studied the rebalancing problem with multiple types of bike (for example, one-seat and two-seat bikes) and resolved their proposed model using a combined hybrid genetic algorithm. In addition, Ho and Szeto [29] looked into the rebalancing operations with the consideration of multiple rebalancing vehicles without permitting shared bikes to be stored at the stations. They also proposed a hybrid large-neighbourhood search algorithm based on several removal and insertion operators to solve the rebalancing model. Using multiple rebalancing vehicles, Bulhões et al. [30] addressed the rebalancing problem by allowing vehicles to visit the bike stations multiple times and proposed an iterative local search metaheuristic to solve the problem. Pal and Zhang [31] considered the rebalancing operations of a free-floating bike-sharing network on the basis of a decomposed network and hybrid nested large-neighbourhood search algorithm. Tang et al. [32] attempted to resolve the repositioning problem with a bi-level programming model in which the upper model determines the number of loaded and unloaded bikes at each station and the lower model optimises the rebalancing route with consideration of the minimum transport cost. Lahoorpoor et al. [33] proposed a cluster method based on the origin and destination of cycling trips and then rebalanced the shared bikes through inter-cluster and intra-cluster methods.

The aforementioned approaches to static rebalancing operations are all carried out from a local perspective, however. That is, in a limited area of the city in which bike-sharing is in service, using single or multiple vehicles to pick up or deliver the shared bikes from one station or parking place to another and allowing single or multiple visits to the same station or parking place. They usually make further assumption that the number of shared bikes in the limited area is fixed. Yet, only from the perspective of the entire city can the total number of shared bikes be regarded as a fixed value within a certain period. Therefore, these studies ignored a very salient problem in the operation and management of a practical bike-sharing network, because they fail to consider the imbalanced distribution of shared bikes among different areas within the city. Nonetheless, the operational scope of bike-sharing networks and the number of shared bikes are both expanding rapidly. For example, the fully serviced area in Beijing encompassing the centre and sub-centre has expanded as far as the Sixth Ring Road, and the number of bikes owned by a bike-sharing company in Beijing exceeded one million by 2018. Moreover, in real-world operation and management, users' cycling trips are not restricted to a fixed area of the city, and the circulation of shared bikes among areas is frequent. Therefore, imbalanced distribution of shared bikes among areas of the city is unavoidable; after each day's operations, the number of bikes may increase in certain areas whilst decreasing in others. If the methods described in the literature were applied to the rebalancing operations of an entire city, the scale and computational cost of the rebalancing problem would be vast. Even if such a model could be formulated for an entire city, finding a satisfactory solution might be computationally intractable due to the practical limitations on computer power. Meanwhile, applying such methods to only a limited area of the city would yield invalid solutions due to the fluctuation of the total number of bikes in such an area. Of course, if the total number of shared bikes in each area could be restored to its original state before the next cycle of rebalancing operation begins, the methods developed in previous studies would be applicable to the whole-city network.

In this paper, to further address the imbalanced distribution problem of shared bikes among different areas, we analyse the area-wise characteristics of imbalanced bike distribution using the real cycling trip data of a bike-sharing company in Beijing. We discover that the rebalancing demand of the static operations in each area is a stochastic variable characterised by multiple scenarios. In addition, from the whole-city perspective, we develop an integer programming model to solve the imbalanced distribution problem among multiple areas and illustrate the inherent properties of the proposed model. Finally, considering a real-world operational scenario, we propose a chance constraint programming model and design eight genetic algorithms with a range of combinations of evolutionary mechanisms. The results demonstrate that the proposed algorithms can effectively solve the static bike rebalancing problem across various areas with a higher solution efficiency than previous genetic algorithms.

The remainder of this paper is organised as follows. Section 2 analyses the characteristic of imbalanced bike distribution based on real cycling trip data. Section 3 formulates the proposed approach for static rebalancing among areas and describes its inherent properties. Section 4 describes the details of combining genetic algorithms with different evolutionary mechanisms to implement the proposed solution mechanism. Section 5 describes the tuning process of the algorithm parameters and provides a case study supported with experimental analysis. In Section 6, the work is summarised with interesting future studies identified.

## 2. Characteristic of imbalanced bike distribution

In this section, we take one of the top two free-floating bike-sharing companies in China as an example to illustrate the imbalanced bike distribution among various areas in Beijing. In a free-floating network, customers use a software app on their mobile phones to locate available shared bikes in their vicinity when they wish to make a trip. At the end of their trip, users can park the bikes in any public parking space and then confirm the trip termination using the app. The data generated in this process are collected by the company. The operational data of such a bike-sharing network mainly relates to the users' cycling trips, including the ID of the borrowed bike, the time interval, and the GPS location of the start and end of each trip, see Table 1. The trip data used in this paper are obtained from the Beijing Transportation Information Center, the department that supervises the bike-sharing companies in Beijing, the data covers around 1.4 million trips per day over the period considered.

Table 1: Introduction to the operation data of bike sharing company

| Bike ID | Rental time | Return time | Rental longitude | Rental latitude | Return longitude | Return latitude |
|---|---|---|---|---|---|---|
| 0106775010 | 20180519073711 | 20180519075809 | 116.57328 | 39.80120 | 116.53508 | 39.80021 |
| 8610103970 | 20180519075649 | 20180519075811 | 116.41114 | 39.92304 | 116.39597 | 39.92213 |
| 0106625457 | 20180519075350 | 20180519075755 | 116.44262 | 39.82746 | 116.44119 | 39.83204 |
| 8630573667 | 20180519075317 | 20180519075649 | 116.49880 | 39.98587 | 116.41145 | 39.99997 |
| 0226622575 | 20180519075335 | 20180519075733 | 116.38359 | 39.89082 | 116.37919 | 39.88839 |

To illustrate the imbalanced bike distribution problem across the entire city, the operational data acquired are pre-processed as follows. First, we restrict the scope of trip data to areas within the Fifth Ring Road of Beijing, which encompass the majority of commuter activities, and divide this entire region into 25 sub-areas of about 5 km × 5 km each, which covers the maximum distance of most cycling trips. Second, because 98.38% of each day's trip data were collected between 06:00 and 00:00, the data for the period from 00:00 to 06:00 is negligible. Therefore, only the trip data from 06:00 to 00:00 on each day are extracted. On this basis, we analyse the trip data of each sub-area in turn, calculating the imbalanced demand of rebalancing operations in each. Specifically, if the start and end points of a trip are located in the same area, this trip has no influence on the number of shared bikes in this area. If the start point of a trip is located in a given area but the end point is not, the original area loses one shared bike. Similarly, if the end point of a trip is located in a given area but the start point is not, the end area gains one shared bike. Thus, after analysing all of the trip data, we can calculate the total numbers of shared bike gained and lost in each area. In this paper, we define the 'variation number' of shared bikes in a given area within the city as follows.

**Definition 1.** The variation number of shared bikes in an area is equal to the difference between the total number of shared bikes gained ($d_i^{increased}$) and the total number of shared bikes lost ($d_i^{decreased}$):

$$d_i^{var} = d_i^{increased} - d_i^{decreased}, i = 1, 2, \cdots, N, \tag{1}$$

where $N$ is the total number of sub-areas considered.

According to Definition 1, we can see that $d_i^{var}$ can take three types of value for each area, including $d_i^{var} > 0$, $d_i^{var} < 0$ and $d_i^{var} = 0$, where $i = 1, 2, \cdots, N$. $d_i^{var} > 0$ indicates that the number of shared bikes in the $i$th area has increased $d_i^{var}$ after the daytime operations. $d_i^{var} < 0$ indicates that the number of shared bikes in the $i$th area has decreased $-d_i^{var}$ after the daytime operations. $d_i^{var} = 0$ indicates that the number of shared bikes in the $i$th area is unchanged after the daytime operations.

For the sake of intuitive illustration, we analyse the cycling trip data of the studied free-floating bike-sharing company on a randomly selected date (18 May 2018) and calculate the variation number of shared bikes in each area ($d_i^{var}, i \in N$). The pattern of the imbalanced distribution is superimposed on the map within the Fifth Ring Road of Beijing in Figure 1. In this figure, the horizontal axis represents longitude and the vertical represents latitude, whereas the colour intensity represents the variation number of shared bikes in each area according to the vertical colour bar on the right. For instance, the variation number of shared bikes in the first area (numbed by 1) is $d_1^{var} = -31$, so the area is coloured light blue, which is the sixth colour in the colour bar from the top.

Figure 1 shows that after the daytime operations, the areas within the Fifth Ring Road differ greatly in the number of available shared bikes, which reflects the city's imbalanced bike distribution, especially on the whole-city scale. Without rebalancing operations, the imbalanced bike distribution will become increasingly serious and eventually threaten user satisfaction with the company. Figure 2 to Figure 13 plot the imbalanced bike distributions for every day

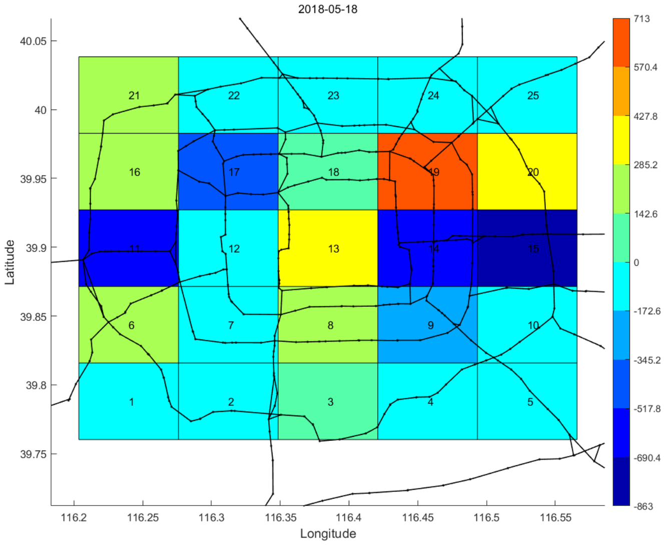

Figure 1: Imbalanced distribution of available bikes between different areas of Beijing on 18 May 2018.

from 18 to 29 May 2018. It can be seen that not only is the bike distribution imbalanced every day, but the variation number of shared bikes in each area is not a constant value; that is, for a given area, the variation number generally takes different values on different days.

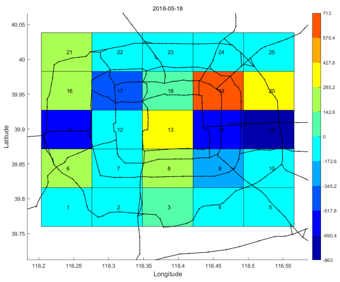

Figure 2: Variation numbers on 2018-05-18.

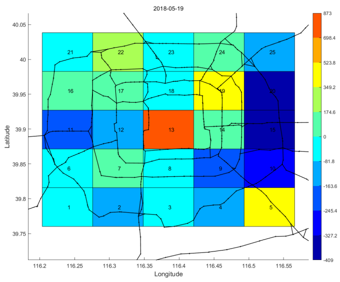

Figure 3: Variation numbers on 2018-05-19.

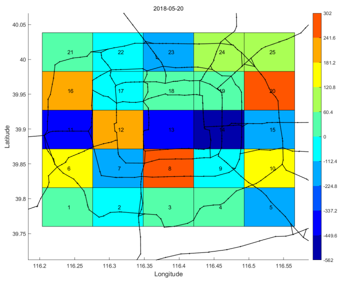

Figure 4: Variation numbers on 2018-05-20.

These findings are intuitively plausible for several reasons. First, because users may refrain from using the network on a certain day as either they are in poor health or they cannot find a working bike available, the total number of trips every day in each area is uncertain. Second, because the choice of commuters to travel by bike is easily affected by the weather, the frequency of cycling trips will be strongly seasonal; in spring and autumn, commuters will be more willing to use shared bikes to complete the beginning or end of their journey, but in summer and winter, they may decide not to cycle at all due to the extreme weather. Finally, commuters may show significantly different patterns in cycling demand on weekdays and weekends, and the characteristics of commuter movement in different areas also have their own behavioural patterns. For example, most commuters need to go to work on weekdays only, which

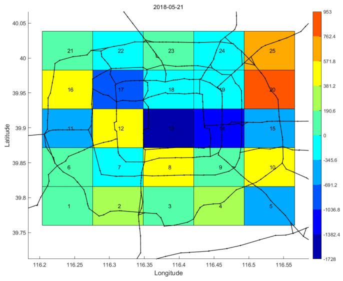

Figure 5: Variation numbers on 2018-05-21.

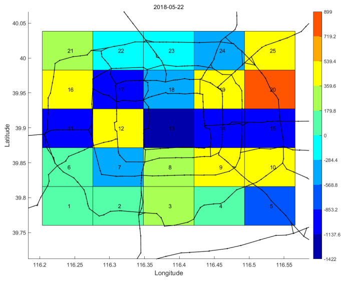

Figure 6: Variation numbers on 2018-05-22.

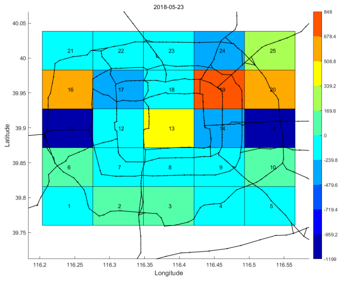

Figure 7: Variation numbers on 2018-05-23.

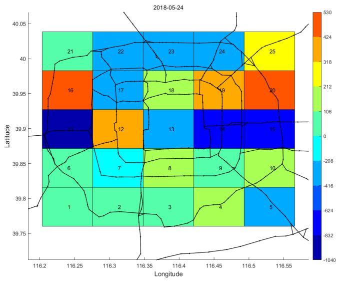

Figure 8: Variation numbers on 2018-05-24.

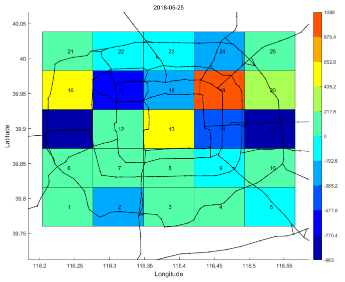

Figure 9: Variation numbers on 2018-05-25.

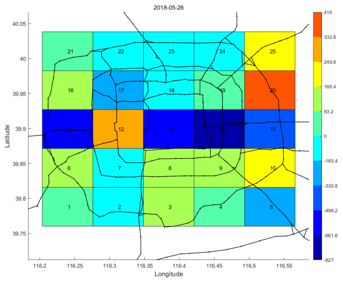

Figure 10: Variation numbers on 2018-05-26.

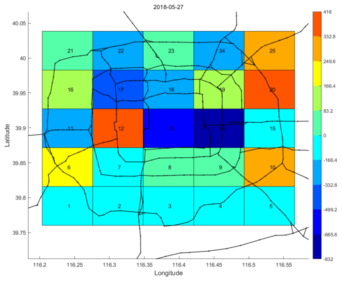

Figure 11: Variation numbers on 2018-05-27.

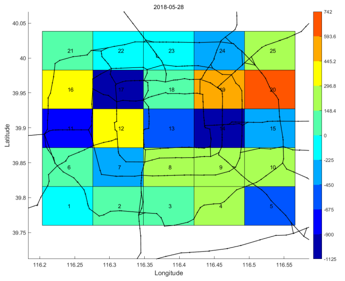

Figure 12: Variation numbers on 2018-05-28.

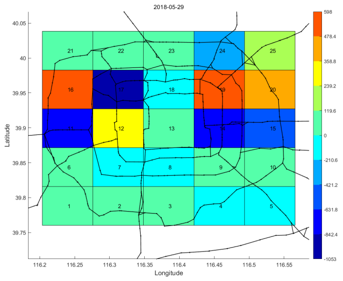

Figure 13: Variation numbers on 2018-05-29.

leads to a huge demand for shared bikes from residential buildings to bus stops or metro/light rail stations, whereas at weekends, shopping, leisure and entertainment venues are more popular destinations, leading to a greater demand for cycling trips in commercial areas. According to the analysis of this company's data, more than 1.4 million cycling trips per day were made on weekdays, but the number at weekends was only around 0.87 million. Therefore, to more accurately capture and describe these sorts of variations in our study, we consider the variation number of shared bikes in each area to be a stochastic value reflecting different scenarios in different time periods. Thus, considering the stochastic rebalancing demand with the existence of multiple scenarios, the aim of this work is to restore the imbalanced bike distribution among different areas in Beijing through static rebalancing operations. In the next section, we describe the construction of the models for rebalancing operations in detail.

## 3. Model construction of bike rebalancing operations

In the following, we use $G = (\mathbb{N}, \mathbb{A})$ to denote the imbalanced bike distribution of the city as a whole, where $\mathbb{N}$ is the collection of areas ($|\mathbb{N}| = N$) and $\mathbb{A}$ is the collection of routes between areas. Suppose that there are $H$ scenarios of imbalanced bike distribution. For each $h = 1, 2, \cdots, H$ and $n \in \mathbb{N}$, we use $d_n^h$ to denote the variation value of area $n$ in scenario $h$. Here $d_n^h > 0$ means that we need to relocate bikes out of this area and $d_n^h < 0$ means that we need to relocate bikes into this area. Note that we should have

$$\sum_{n \in \mathbb{N}} d_n^h = 0, \ \forall h = 1, 2, \cdots, H. \tag{2}$$

We know from the conversations with porters of a bike sharing company that lorries and electric tricycles are two common tools used by the bike sharing company studied. To efficiently conduct static rebalancing, we use a combination of lorries and electric tricycles to rebalance the shared bikes among different areas in this study. Here suppose that each lorry has a capacity of $\alpha$ and each electric tricycle has capacity $\beta$. Without loss of generality, we assume that $\beta = 1$ as it is the relative capacity between the two types of bike rebalancing vehicle that is of interest. Note that due to the substantial fixed costs of possessing own rebalancing vehicles, bike-sharing companies do not normally purchase them outright but instead hire them. Rebalancing lorries have a larger capacity and are the mainstay of rebalancing operations, but they have a higher rental cost. Bike-sharing companies typically sign a contract with the vehicle rental companies at the beginning of a certain operational period (usually half a year or more) to determine the number of lorries needed and the corresponding rental cost, which cannot be subsequently changed during the operational period once the contract is established. However, due to the stochasticity of rebalancing demand, the company cannot be certain regarding whether the number of lorries rented in advance will meet the actual rebalancing demand, so they will rent electric tricycles to make up for any shortfall. In general, the rental cost of electric tricycles is sensitive to the time-varying (i.e., seasonal) price of electricity and the amount of supply in the rental market. On the basis of these considerations, we assume that the cost for each lorry trip is $c_t$ and that for each electric tricycle trip is $c_e$. We also assume that the time-varying price coefficient is $p_h$ under scenario $h$, which is a variable for electric tricycles and preset to 1 for lorries (again, only the relevant cost is of interest between the two types of rebalancing vehicle).

Recall that the present research goal is to seek a balanced distribution of shared bikes throughout the city. This rebalancing operation requires the transfer of shared bikes between different areas of the city. Picking up and delivering bikes with the rebalancing vehicles takes time, as does driving the rebalancing vehicles from one area to another. Thus, we assume that each rebalancing vehicle only performs one operation per night in this study, which is typically the case in the real setting. The decision variables considered can therefore be represented as follows:

- $y_{ij}$: Number of lorry trips at arc $(i, j) \in \mathbb{A}$, which must be determined at the beginning of each period because a long-term contract is required for hiring a lorry. Denote $Y = \{y_{ij} \mid i, j = 1, 2, \cdots, N\}$.
- $z_{ij}^h$: Number of electric tricycle trips at arc $(i, j) \in \mathbb{A}$ in scenario $h$, which is determined at the time of rental because no contract is required for hiring an electric tricycle. Denote $Z = \{z_{ij}^h \mid i, j = 1, 2, \cdots, N, h = 1, 2, \cdots, H\}$.
- $x_{ij}^h$: Number of bikes moved from node (or area) $i$ to node $j$ in scenario $h$. Denote $X = \{x_{ij}^h \mid i, j = 1, 2, \cdots, N, h = 1, 2, \cdots, H\}$.

From these, the basic static rebalancing model can be represented as below:

$$(P_0) \quad \begin{cases} \min\limits_{X,Y,Z} & \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} c_t y_{ij} + \sum\limits_{h=1}^{H}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} p_h c_e z_{ij}^h \\ \text{s.t.} & \sum\limits_{j=1}^{N} x_{ij}^h = \max\left\{d_i^h, 0\right\},\ i = 1, 2, \cdots, N,\ h = 1, 2, \cdots, H \\ & \sum\limits_{i=1}^{N} x_{ij}^h = \max\left\{-d_j^h, 0\right\},\ j = 1, 2, \cdots, N,\ h = 1, 2, \cdots, H \\ & z_{ij}^h = \max\left\{x_{ij}^h - \alpha y_{ij}, 0\right\},\ i, j = 1, 2, \cdots, N,\ h = 1, 2, \cdots, H \\ & x_{ij}^h, y_{ij}, z_{ij}^h \in Z^+,\ i, j = 1, 2, \cdots, N,\ h = 1, 2, \cdots, H. \end{cases} \tag{3}$$

The objective of this model is to minimise the total cost of using the lorries and electric tricycles for rebalancing. The first two constraints are aimed at maintaining the balance between supply and demand of bikes in each area under each scenario. The third constraint ensures that the number of shared bikes carried by electric tricycles is equal to the number remaining after all relocations carried out by all lorries. The last constraint guarantees that the decision variables are (encoded as) non-negative integers in the real-world operations.

**Theorem 1.** *The basic model has the following integer linear equivalent form*

$$(P_1) \quad \begin{cases} \min\limits_{X,Y,Z} & \sum_{i=1}^{N}\sum_{j=1}^{N} c_t y_{ij} + \sum_{h=1}^{H}\sum_{i=1}^{N}\sum_{j=1}^{N} p_h c_e z_{ij}^h \\ s.t. & \sum_{j=1}^{N} x_{ij}^h = \max\left\{d_i^h, 0\right\}, \ i = 1,2,\cdots,N, \ h = 1,2,\cdots,H \\ & \sum_{i=1}^{N} x_{ij}^h = \max\left\{-d_j^h, 0\right\}, \ j = 1,2,\cdots,N, \ h = 1,2,\cdots,H \\ & x_{ij}^h \le \alpha y_{ij} + z_{ij}^h, \ i,j = 1,2,\cdots,N, \ h = 1,2,\cdots,H \\ & x_{ij}^h, y_{ij}, z_{ij}^h \in Z^+, \ i,j = 1,2,\cdots,N, \ h = 1,2,\cdots,H. \end{cases} \tag{4}$$

*Proof.* If $(X, Y, Z)$ is the optimal solution of the basic model $(P_0)$, we have $Z = \max\{X - \alpha Y, 0\}$, which implies that $X \le \alpha Y + Z$ and $Z \ge 0$ such that $(X, Y, Z)$ is a feasible solution of model $(P_1)$. Conversely, if $(X, Y, Z)$ is the optimal solution of model $(P_1)$, we have $X \le \alpha Y + Z$ and $Z \ge 0$, which implies that $Z \ge \max\{X - \alpha Y, 0\}$. Furthermore, because the objective is to minimise a linear function of $Z$ with positive coefficients, the equality holds such that $(X, Y, Z)$ is a feasible solution of the basic model $(P_0)$. The proof is completed. □

**Theorem 2.** *If $c_t > \alpha \times c_e$, the optimal solution $(X, Y, Z)$ satisfies that $X = Z$ and $Y = 0$.*

*Proof.* Let $(X, Y, Z)$ be an optimal solution. If $Y > 0$ such that $y_{ij} \ge 0$ for $i, j = 1, 2, \cdots, N$, there is at least one pair of indices $(i, j)$ with $y_{ij} > 0$, we define a new feasible solution $\overline{X} = X$, $\overline{Y} = 0$, $\overline{Z} = \alpha Y + Z$. Because $c_t > \alpha \times c_e$, we have

$$F\left(\overline{X}, \overline{Y}, \overline{Z}\right) = \sum_{h=1}^{H}\sum_{i=1}^{N}\sum_{j=1}^{N} p_h c_e \left(\alpha y_{ij} + z_{ij}^h\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha c_e y_{ij} + \sum_{h=1}^{H}\sum_{i=1}^{N}\sum_{j=1}^{N} p_h c_e z_{ij}^h < F(X, Y, Z),$$

which contradicts the optimality of $(X, Y, Z)$. Furthermore, because the objective is to minimise a linear function of $Z$ with positive coefficients under constraint $X \le Z$, we have $X = Z$. The proof is completed. □

The above theorem indicates that if the rental cost of one lorry is higher than that of $\alpha$ electric tricycles, the optimal solution is to rent the electric tricycles. In this case, the main rebalancing model $(P_1)$ degenerates to the following transport model $(P_2)$

$$(P_2)\quad\begin{cases}\min\limits_{X,Y,Z} & \sum\limits_{h=1}^{H}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} p_h c_e x_{ij}^h\\ \text{s.t.} & \sum\limits_{j=1}^{N} x_{ij}^h = \max\left\{d_i^h, 0\right\},\ i=1,2,\cdots,N,\ h=1,2,\cdots,H\\ & \sum\limits_{i=1}^{N} x_{ij}^h = \max\left\{-d_j^h, 0\right\},\ j=1,2,\cdots,N,\ h=1,2,\cdots,H\\ & x_{ij}^h \in Z^+,,\ i,j=1,2,\cdots,N,\ h=1,2,\cdots,H.\end{cases}\tag{5}$$

Note that the number of decision variables is $(2H+1)\times N^2$ and the number of constraints is $H\times(N^2+2N)$. For large balancing problems with a higher value of $N$, we must perform a suitable decomposition into sub-models to speed up the solution procedure. The first sub-model $(P_3)$ below finds the optimal scheduling strategy for lorries and electric tricycles:

$$(P_3)\quad\begin{cases}\min\limits_{Y,Z} & \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} c_t y_{ij} + \sum\limits_{h=1}^{H}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} p_h c_e z_{ij}^h\\ \text{s.t.} & \sum\limits_{j=1}^{N}\left(\alpha y_{ij} + z_{ij}^h\right) \ge \max\left\{d_i^h, 0\right\},\ i=1,2,\cdots,N,\ h=1,2,\cdots,H\\ & \sum\limits_{i=1}^{N}\left(\alpha y_{ij} + z_{ij}^h\right) \ge \max\left\{-d_j^h, 0\right\},\ j=1,2,\cdots,N,\ h=1,2,\cdots,H\\ & y_{ij}, z_{ij}^h \in Z^+,\ i,j=1,2,\cdots,N,\ h=1,2,\cdots,H.\end{cases}\tag{6}$$

The second sub-model $(P_4)$ finds a feasible balancing strategy $X$ with a given solution $Y, Z$:

$$(P_4)\quad\begin{cases}\sum\limits_{j=1}^{N} x_{ij}^h = \max\left\{d_i^h, 0\right\},\ i=1,2,\cdots,N,\ h=1,2,\cdots,H\\ \sum\limits_{i=1}^{N} x_{ij}^h = \max\left\{-d_j^h, 0\right\},\ j=1,2,\cdots,N,\ h=1,2,\cdots,H\\ x_{ij}^h \le \alpha y_{ij} + z_{ij}^h,\ i,j=1,2,\cdots,N,\ h=1,2,\cdots,H\\ x_{ij}^h \in Z^+,\ i,j=1,2,\cdots,N,\ h=1,2,\cdots,H.\end{cases}\tag{7}$$

231 **Theorem 3.** *If $(X, Y, Z)$ is a feasible solution of model $(P_0)$, then $(Y, Z)$ is a feasible solution of sub-model $(P_3)$.*

*Proof.* For each $i = 1, 2, \cdots, N$ and $h = 1, 2, \cdots, H$, we have

$$\sum_{j=1}^{N} \left(\alpha y_{ij} + z_{ij}^{h}\right) \geq \sum_{j=1}^{N} x_{ij}^{h} = \max\left\{d_i^h, 0\right\},$$

and for each $j = 1, 2, \cdots, N$ and $h = 1, 2, \cdots, H$, we have

$$\sum_{i=1}^{N} \left(\alpha y_{ij} + z_{ij}^{h}\right) \geq \sum_{i=1}^{N} x_{ij}^{h} = \max\left\{-d_j^h, 0\right\},$$

which implies that $(Y, Z)$ is a feasible solution of sub-model (P$_3$). The proof is completed. □

**Theorem 4.** *If $(Y, Z)$ is an optimal solution of sub-model (P$_3$) and X is a feasible solution of sub-model (P$_4$) with $(Y, Z)$, then $(X, Y, Z)$ is an optimal solution of model (P$_0$).*

*Proof.* Let $(X, Y, Z)$ denote an optimal solution of model (P$_0$). If there would be another optimal solution $(X_1, Y_1, Z_1)$, it would follow from Theorem 3 that $(Y_1, Z_1)$ would be a feasible solution of model (P$_3$) with a lower objective value. However, this contradicts the optimality of $(Y, Z)$. The proof is completed. □

**Theorem 5.** *If $(X, Y, Z)$ is an optimal solution of model (P$_0$), then $(Y, Z)$ is an optimal solution of model (P$_3$) and X is a feasible solution of sub-model (P$_4$) with $(Y, Z)$.*

*Proof.* The feasibility of $X$ is trivial. Now we prove the optimality on $(Y, Z)$. According to Theorem 3, $(Y, Z)$ is a feasible solution of sub-model (P$_3$). If there would be another optimal solution $(Y_1, Z_1)$ with lower objective value, denote $X_1$ as the corresponding feasible solution of sub-model (P$_4$), then it would follow from Theorem 4 that $(X_1, Y_1, Z_1)$ would be the optimal solution of model (P$_0$). This contradicts to the optimality of $(X, Y, Z)$. The proof is completed. □

As previously discussed, in practical operation and management of a bike-sharing network, the serving company needs to consider the stochastic fluctuation of rebalancing demand in each area under time-varying scenarios. Therefore, by introducing the concept of service satisfaction level into the previous rebalancing model (P$_3$), we propose a chance constraint rebalancing model (P$_5$) as follows.

$$(P_5) \quad \left\{\begin{array}{ll} \min\limits_{Y,Z} & \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} c_t y_{ij} + \sum\limits_{h=1}^{H}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} p_h c_e z_{ij}^{h} \\ \text{s.t.} & \Pr\left\{\sum\limits_{j=1}^{N}\left(\alpha y_{ij} + z_{ij}^{h}\right) \geq \max\left\{d_i^h, 0\right\}\right\} \geq \gamma_1, \ i = 1, 2, \cdots, N, \ h = 1, 2, \cdots, H \\ & \Pr\left\{\sum\limits_{i=1}^{N}\left(\alpha y_{ij} + z_{ij}^{h}\right) \geq \max\left\{-d_j^h, 0\right\}\right\} \geq \gamma_2, \ j = 1, 2, \cdots, N, \ h = 1, 2, \cdots, H \\ & y_{ij}, z_{ij}^{h} \in Z^{+}, \ i, j = 1, 2, \cdots, N, \ h = 1, 2, \cdots, H. \end{array}\right. \tag{8}$$

This model is also set to minimise the total cost of operating multiple rebalancing vehicles. The first two constraints ensure that the capacity of the scheduling strategy of lorries and electric tricycles meets the $\gamma$ level under the stochastic rebalancing demand in each area and scenario, in which $\gamma_1$ is the satisfaction level of the first constraint and $\gamma_2$ is that of the second. In real-world practice, the managers of the bike sharing network have to adjust the value of satisfaction level $\gamma$ according to the current status and business goals of the company. The last constraint simply guarantees that the decision variables are (again, encoded as) non-negative integers.

## 4. Genetic algorithms based on variable mutation operator with different evolutionary mechanisms

We have introduced a chance constraint integer programming model ($P_5$) as described in Section 3, which can be rather complex to resolve in real-world settings. Heuristic algorithms have demonstrated their strengths over deterministic methods in dealing with large-scale stochastic programming models [34, 35, 36, 37]. Therefore, we employ genetic algorithms that are based on a variable mutation operator with different evolutionary mechanisms to solve this complicated problem. In particular, we utilise multiple squares to form the shape of the chromosome in the genetic algorithms and propose different mechanisms for the genetic processes of crossover, mutation and iterative update. The basic framework of the genetic algorithm-based approach is summarised in Table 2, implementation details of this framework are explained next.

Table 2: Framework of genetic algorithm-based approach

**Step 1:** Generate a population with $K$ feasible chromosomes;
**Step 2:** Calculate the fitness value of each chromosome;
**Step 3:** Select two chromosomes randomly to implement the crossover procedure by roulette;
**Step 4:** Select a chromosome randomly to implement the mutation procedure;
**Step 5:** Repeat step 2 to step 4 until the termination condition is reached;
**Step 6:** Calculate the optimal value and obtain the optimal results.

### *4.1. Generation of chromosome population*

The first step in a genetic algorithm is for the encoding of the chromosomes. In general, the chromosomes may be encoded in a variety of structures, including binary, floating-point and symbolic. In this study, considering the characteristics of the proposed model ($P_5$), we develop a hierarchical square chromosome encoding method, in which the chromosomes are composed of multiple square matrices.

Recall that in model ($P_5$), the decision variables consist of $y_{ij}$ and $z_{ij}^h$, where $i, j = 1, 2, \cdots, N$, $h = 1, 2, \cdots, H$. Thus, if $y_{ij}$ is a feasible solution of the model, its structure will consist of $N$ rows and $N$ columns. Similarly, if $z_{ij}^h$ is a feasible solution, it will be made up of $H$ squares, each consisting of $N$ rows and $N$ columns. Therefore, we build the structure of the chromosome with $H+1$ squares. Figure 14 is a schematic diagram of the chromosome structure, in which the first square is the variable of the rebalancing lorries $y_{ij}$ and the following $H$ squares are the variables of the rebalancing electric tricycles $z_{ij}^h$ under different scenarios; for example, in the second scenario ($h = 2$), the results of the model will be $y_{ij}$ and $z^2_{ij}$. The proposed genetic algorithm proceeds through the following steps based on the above chromosome structure.

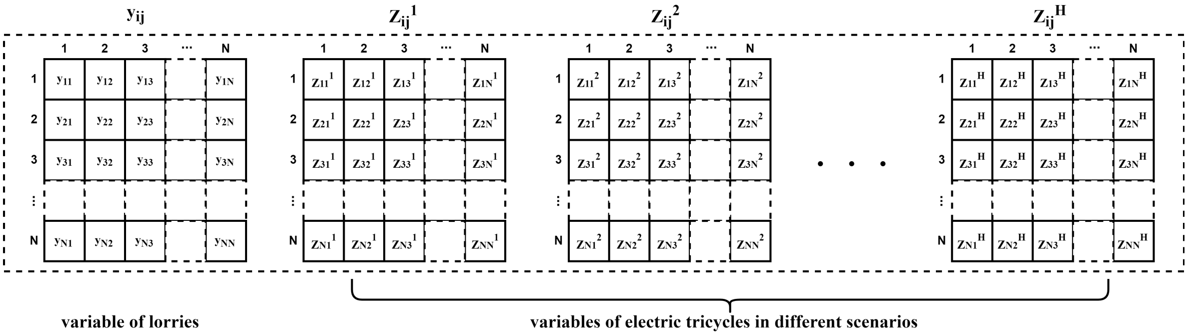

Figure 14: Chromosome structure of decision variables.

A population of such encoded chromosomes is generated via the application of stochastic simulation. To facilitate understanding, we briefly explain the application process of stochastic simulation (as further details are beyond of the scope of this paper). In a chance constraint programming model, a satisfaction level indicator $r$ is typically used to characterise the results. To judge whether the solution expressed by a certain chromosome meets the chance constraint condition during the evolution of the genetic algorithm, the chromosome is subjected to the constraints represented by $N$ random variables. When $r \times N$ constraints satisfy the given validation criteria, we consider that the chromosome meets the constraints; otherwise, it does not. Repeating this process yields a feasible solution of the model. As such, by repeating the above process $K$ times, we can obtain a population containing $K$ feasible chromosomes. Note that readers who wish to further familiarise themselves with stochastic simulation-based genetic algorithms can refer to the work presented in [38] and [34].

### 4.2. *Crossover procedure*

Chromosome crossover is an important part of a genetic algorithm. A well-designed crossover mechanism enables the solution space of large-scale models to be searched more efficiently, greatly reducing the computational time required for hunting the optimal solution. Considering the need both for the ability of the algorithm to jump out of local optimality and for the efficiency of the local search process, two crossover mechanisms are proposed in this study, namely: point crossover mechanism (PCM) and block crossover mechanism (BCM).

#### 4.2.1. *Point crossover mechanism (PCM)*

In each iteration of crossover, we select several points randomly in each layer of the chromosome and exchange these points from one chromosome to another. If the chromosomes that have undergone the point exchange procedure all satisfy the constraints of the model, the crossover procedure of this iteration is completed; otherwise, we repeat the PCM. Figure 15 is a schematic diagram of the PCM in the case where two points are exchanged.

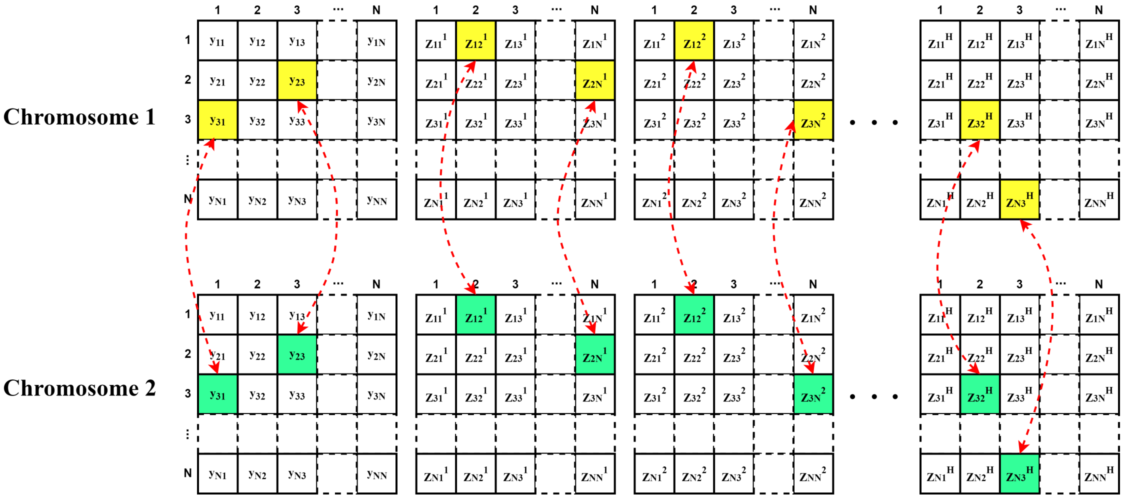

Figure 15: Point crossover mechanism.

### *4.2.2. Block crossover mechanism (BCM)*

Each exchange of gene fragments by PCM only accounts for a relatively small number of genes. To expand the search ability of the genetic algorithm, we can also use BCM. In this process, having selected the chromosomes between which we wish to exchange genes, we randomly select a rectangular block in each layer of each chromosome, and then exchange the genes between the rectangular blocks of two chromosomes. If the chromosomes that have undergone the BCM all satisfy the constraints of the model, the crossover procedure of this iteration is completed; otherwise, we repeat the BCM. Figure 16 shows a schematic diagram of BCM.

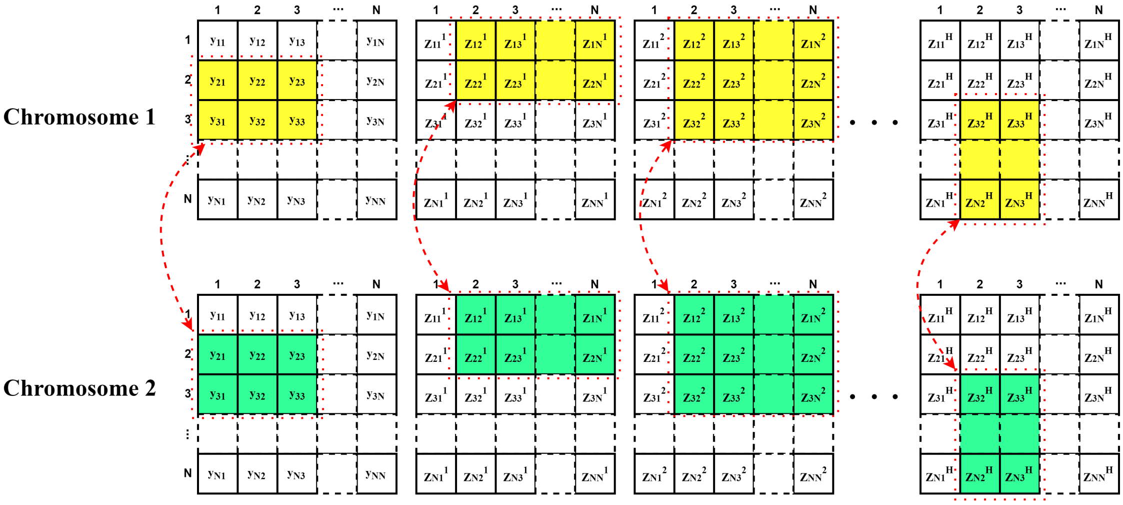

Figure 16: Block crossover mechanism.

In implementation, a specific control is exercised such that at initial stages, if PCM or BCM fails to satisfy the constraints in an iteration, the crossover procedure is repeated, and that if a satisfactory solution is yet not found after conducting the process for a fixed number of times (five in the present study), the algorithm will exit from the crossover iteration and move to the next procedure.

### *4.3. Chromosome update mechanism*

The optimal solution of the chromosomes in the given population will change after running the crossover procedure; that is, sometimes the solution will be better than before, sometimes worse. Most commonly, the solution is directly updated with the iterative solution after each iteration, regardless of whether the optimality of the solution improves, which is known as the direct update mechanism (DUM) in the literature. However, this may cause the solution to converge too slowly. To improve upon this update mechanism, we set the algorithm only to update the chromosomes as the value of the iterative solution is better than the previous iteration; otherwise, we re-execute the crossover operation. We refer to this as improved update mechanism (IUM) hereafter. These two mechanisms have their own strengths: DUM increases the diversity of the solution search, while IUM accelerates the convergence of the algorithm.

### *4.4. Mutation procedure*

Mutation is another essential procedure of genetic algorithms that helps to improve the results within local feasible areas. In this study, as consider the mutation operator as a variable in the genetic algorithm, by introducing two mutation mechanisms to enhance its local search power.

In particular, when running each iteration of mutation, we (i) select a few points of a certain chromosome in each layer randomly, and then (ii) subtract a random number of points given by the positive integer $\delta_k^t$, $k = 1, 2, \cdots, H + 1$, where $t$ represents the total number of points selected in step (i). At each selected location, $\delta_k^t$ does not exceed the number of selected points $t$. If the mutant chromosome satisfies the constraints of the model, the mutation procedure is completed. However, if the constraints are not satisfied, we replace the number of subtracted points $\delta$ by half of itself, i.e., $\delta_{new}^t = [\delta_{old}/2]$, where $[\cdot]$ represents the floor integer transformation of the variable in the square brackets, and repeat the deletion procedure until all constraints are satisfied. Because we subtract $\delta_k^t$ from its original value every time, we call this mutation process directional mutation mechanism (DMM). Although DMM is of the ability of faster convergence, it may be easy to fall into a local optimal solution. Therefore, we propose another mutation mechanism, in which every time we mutate a chromosome, we add $\delta_k^t$ to the original value with a probability of 50%, or subtract $\delta_k^t$ from the original value, again with a probability of 50%. We call this non-directional mutation mechanism (NMM).

If both mutation procedures fail to find a feasible solution, we narrow the range of gene mutation by selecting only one point randomly in each layer of the chromosome. If these mutation procedures still fail to satisfy the constraints, we repeat the process until a feasible solution of the model is found. Note that in the following investigations, most

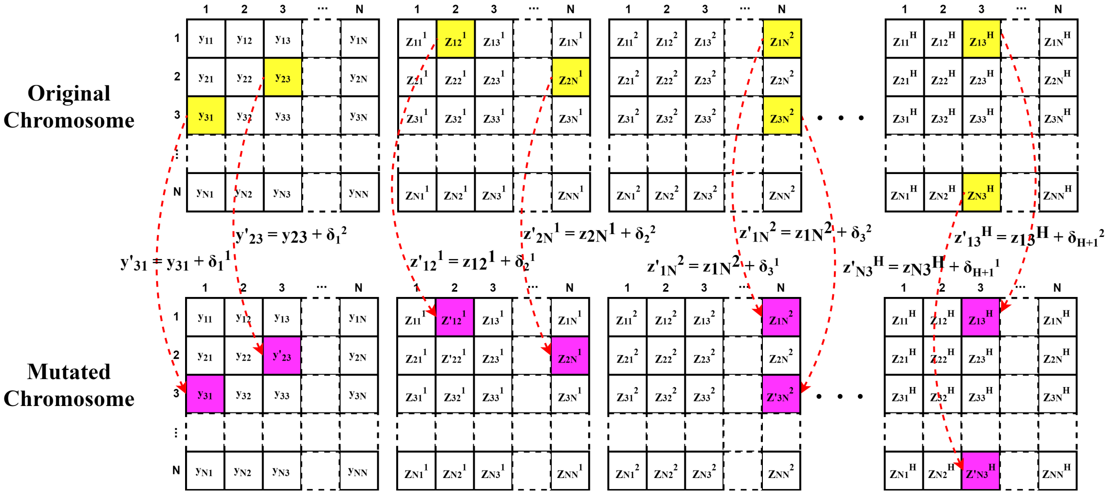

Figure 17: Two-point mutation process.

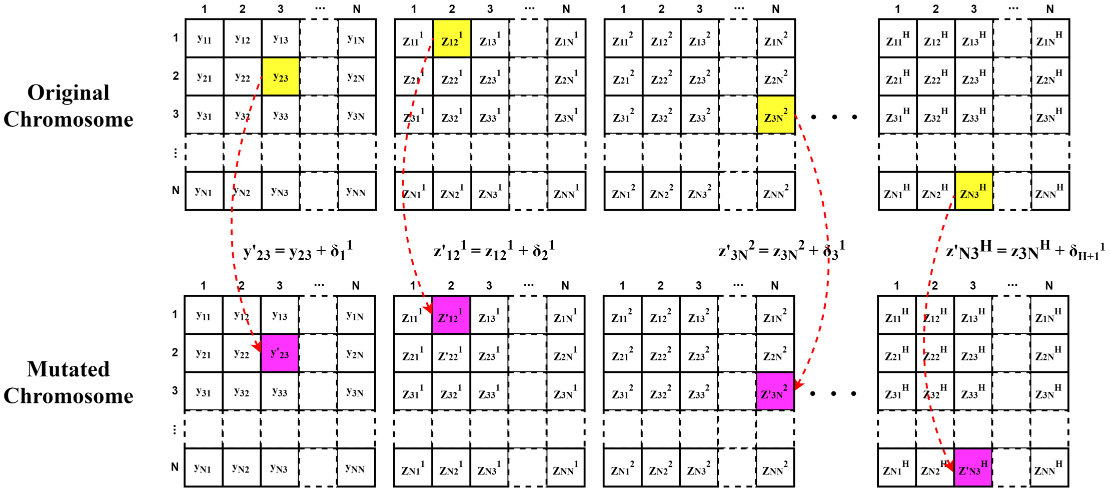

Figure 18: Single-point mutation process.

mutation operations are implemented using two points or a single point in each layer of the chromosome. Figures 17 and 18 illustrate the schematic diagrams of the two-point and single-point mutation mechanism, respectively.

So far we have proposed two crossover mechanisms (PCM and BCM), two update mechanisms (DUM and IUM) and two mutation mechanisms (NMM and DMM). By combining these mechanisms, we can obtain eight genetic algorithms with different evolutionary characteristic, as listed in Table 3. The resulting algorithms each have their own strengths, though it is difficult to generally predict which is most suitable for the rebalancing operations of bike sharing networks. We verify these algorithms in terms of their potential suitability with regard a realistic network in the next section.

Table 3: Eight genetic algorithms with different evolution mechanisms

| Name | Combination of mechanisms |
|---|---|
| GA1 | PCM + DUM + NMM |
| GA2 | PCM + DUM + DMM |
| GA3 | PCM + IUM + NMM |
| GA4 | PCM + IUM + DMM |
| GA5 | BCM + DUM + NMM |
| GA6 | BCM + DUM + DMM |
| GA7 | BCM + IUM + NMM |
| GA8 | BCM + IUM + DMM |

## 5. Parameter tuning and case study

In this section, we first introduce the process of adjusting the parameters of the genetic algorithms to run and then, use a real-world example to identify which genetic algorithm specification may be the most suitable for static bike rebalancing operations, taking the whole city of Beijing as the case-study area.

### *5.1. Parameter tuning for proposed algorithms*

The parameters that need to be tuned in a genetic algorithm include the number of iterations $T$, the number of chromosomes in the population $K$, the crossover rate $Cr$ and the mutation rate $Cm$. Below we use GA1 as an example to illustrate the adjustment process of these parameters. Note that all implemented systems are run on a Windoes 10 personal computer, with an i7-8550u CPU and 32 GB memory, using the software Matlab, version 2019a.

The first parameter to tune is the appropriate number of iterations. We consider the optimal value to have been reached when the population no longer changes from one iteration to the next. Initially, the maximum number of iterations allowed is set to 5000 while recording the best objective value obtained so far over the iterations, as shown in Figure 19. From the figure we can see that as GA1 continues to iterate, the improvement gained from fine-tuning the iteration number gradually slows down. We observe that the best value is essentially unchanged after the number of iterations exceeds 2500, so for GA1 we set $T$=2500 as the number of iterations in the following research.

Then, using 2500 iterations, we re-run GA1 with the number of populations set to 10, 30, 50, 70 and 90. The obtained results are compared in Figure 20. From this figure we can see that when the number of populations is 10, GA1 converges very slowly, and as the number of populations is increased to 30, 50, 70 and 90, the convergence accelerates. However, the price paid (the total computation time of 2500 iterations and the unit computation time of each iteration) also significantly increases, as shown in Table 4. Because the convergence speeds in the last four cases (where the population size, popsize = 30, 50, 70, 90) are all fairly similar when the algorithm is iterated to 2500 generations, it is therefore appropriate to set popsize = 30 in GA1, to have a trade off between effectiveness and efficiency.

Finally, we move to determine the appropriate values of crossover rate and mutation rate, through two steps while fixing $T = 2500$ and $K = 30$. In the first step, we vary both crossover rate and mutation rate, allowing each to

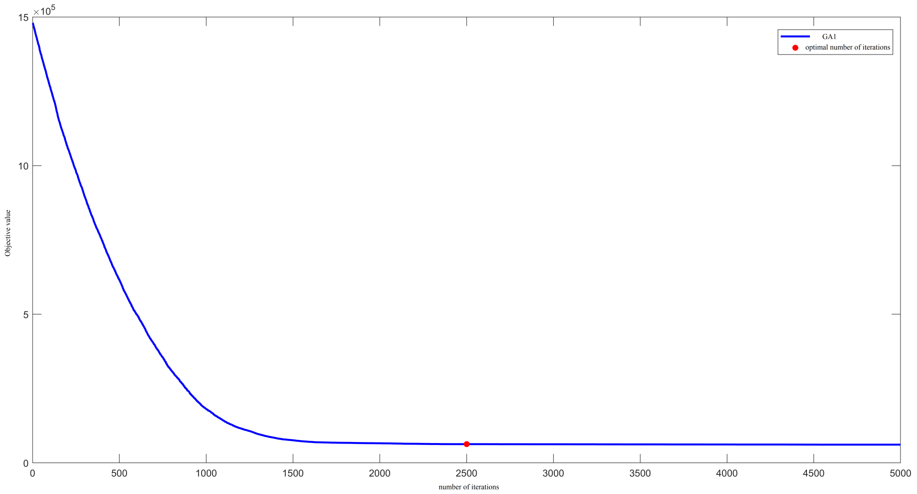

Figure 19: Determination of optimal number of iterations.

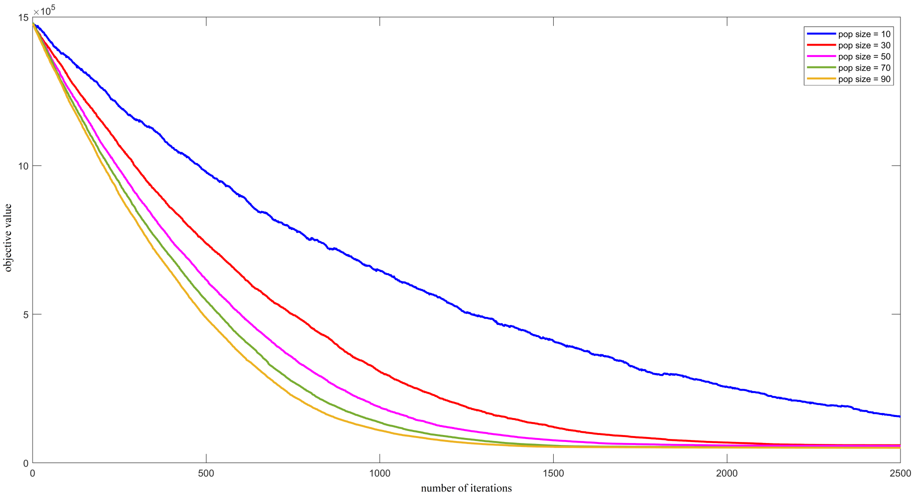

Figure 20: Selection of appropriate population size.

Table 4: Iteration time given different popsizes

| popsize<br>time(s) | 10 | 30 | 50 | 70 | 90 |
|---|---|---|---|---|---|
| total computation time | 961.9 | **3,944.6** | 20,579.3 | 22,672.4 | 29,846.0 |
| unit computation time | 0.385 | **1.578** | 8.232 | 9.069 | 11.938 |

independently take values of 0.1, 0.3, 0.5, 0.7 and 0.9. For example, when we set the parameter combination of GA1 to $Cr$ = 0.1 and $Cm$ = 0.1 (with $T$ = 2500 and $K$ = 30), we obtain the objective value of the model being 223,836, as listed in Table 5. Then we set the crossover rate and mutation rate to each of the other possible combinations and recalculate the objective values, which are likewise listed in Table 5. From this table it can be seen that the best objective value amongst them is reached when $Cr = 0.7$ and $Cm = 0.9$, through this first step. Then, based on this parameter combination, we narrow the interval of the parameters, and then recalculate the objective values under various parameter combinations to obtain the values as recorded in Table 6. Now we can see that the objective value is still best when $Cr = 0.7$ and $Cm = 0.9$. Therefore, through the above two-step parameter tuning, we obtain the optimal parameter combination of GA1 as $T = 2500$, $K = 30$, $Cr = 0.7$, and $Cm = 0.9$.

Table 5: First tuning step of *Cr* and *Cm*

| Objective value | | Crossover rate | | | | |
|---|---|---|---|---|---|---|
| | | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| Mutation rate | 0.1 | 223,836 | 75,124 | 75,770.5 | 70855 | 64,874 |
| | 0.3 | 135,460.5 | 76,793 | 57,551 | 69,092.5 | 62,005.5 |
| | 0.5 | 87,090 | 77,165 | 66,642.5 | 80,029 | 78,346 |
| | 0.7 | 80,312.5 | 72,943.5 | 59,880 | 75,072.5 | 52,003.5 |
| | 0.9 | 83,605 | 65,754 | 67,587 | **42,115.5** | 52,690.5 |

Table 6: Second tuning step of *Cr* and *Cm*

| Objective value | | Crossover rate | | | | |
|---|---|---|---|---|---|---|
| | | 0.65 | 0.675 | 0.7 | 0.725 | 0.75 |
| Mutation rate | 0.85 | 51,355.5 | 62,748.5 | 48,728 | 47277 | 69788.5 |
| | 0.875 | 50,383 | 52,273 | 44,062 | 54878 | 61167.5 |
| | 0.9 | 60,423.5 | 56,437.5 | **42,115.5** | 48902.5 | 61985.5 |
| | 0.925 | 64,443.5 | 54,909.5 | 95,380.5 | 65006 | 47397.5 |
| | 0.95 | 62,981.5 | 88,048 | 60388 | 83662 | 54404.5 |

Now, we use the same method to tune the optimal parameter combinations for the other seven genetic algorithms, with results recorded in Table 7. Having established the optimal parameters of the eight algorithms, we next determine which algorithm performs the best in coping with a real-world problem of bike-sharing rebalancing.

Table 7: Optimised parameters of eight genetic algorithms

| Name | $T$ | $K$ | $Cr$ | $Cm$ |
|---|---|---|---|---|
| GA1 | 2500 | 30 | 0.7 | 0.9 |
| GA2 | 2500 | 30 | 0.65 | 0.85 |
| GA3 | 2500 | 30 | 0.6 | 0.8 |
| GA4 | 2500 | 30 | 0.7 | 0.75 |
| GA5 | 2500 | 30 | 0.8 | 0.7 |
| GA6 | 2500 | 30 | 0.75 | 0.8 |
| GA7 | 2500 | 30 | 0.7 | 0.85 |
| GA8 | 2500 | 30 | 0.75 | 0.85 |

### 5.2. *Case study*

In this section, we compare the proposed genetic algorithms themselves, based on the use of the optimal parameters of Table 7, and also with an existing popular software package that represents the state-of-the-art deterministic algorithms in solving a real-world bike-sharing rebalancing problem. The operational situation and data sources of the bike-sharing company are as described in Section 2. The data used in this case study covers 74 days of the company's operation in 2018, including 51 weekdays and 23 weekends days. We divide the whole of Beijing city into 29 areas, of which 25 are shown in Figure 1 in Section 2; the other four are the parts that lie beyond that map to the east, south, west and north, respectively. We utilise the daily cycling trip data to calculate the rebalancing demands in these 29 areas and divide the rebalancing demand data of the 74 days into two groups: weekdays and weekends.

As an alternative to our chance constrain planning model (P5), namely model (P3), may be implemented using optimisation software packages such as Cplex, Gurobi and Mosek. Based on availability, we use Gurobi 9.0.0 as an example to show why this type of optimisation software is not applicable in realistic bike-sharing operations. Since any lease contract for lorries must be signed in advance, we can obtain an exact description for the input to model (P3) using historical rebalancing demand data. In particular, to avoid potential bias, we take the average value of rebalancing demand as the input and use Gurobi to obtain the number of lorries and electric tricycles required, for comparison. After that, we feed the computed results back to model (P3) to calculate the satisfaction level of those constraints involved, per day. Figure 21 shows the resulting daily satisfaction level. We can see that based on the average historical rebalancing demand, the results obtained from model (P3) fluctuate greatly from day to day, in terms of the satisfaction level. The average value of the 74 days' satisfaction level is a mere 66.6%, which is far below the company's target satisfaction level of 80%. Thus, whilst the optimisation software may obtain the optimal solution of a deterministic model, such a solution is not suitable for scheduling the real-world rebalancing operations, especially when the rebalancing demand is a stochastic variable with multi-scenario characteristics.

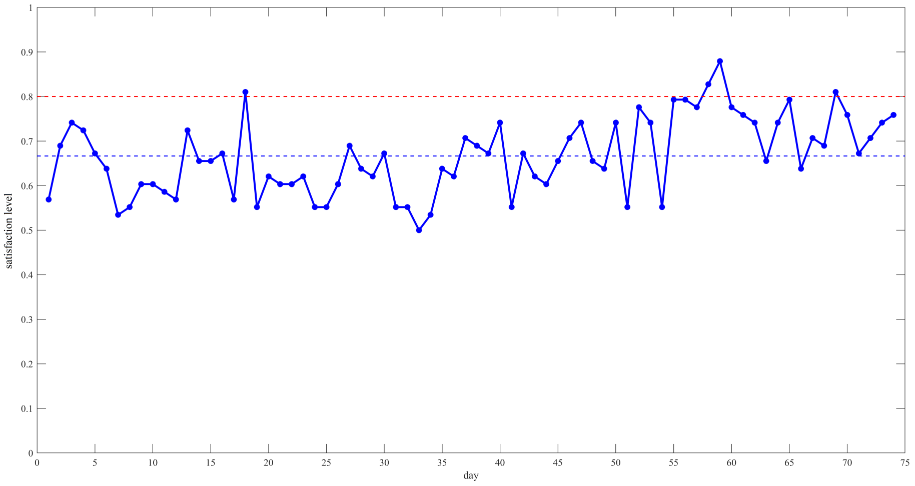

Figure 21: Satisfaction level of solutions calculated by Gurobi software package.

In the following, we experimentally investigate the effects of applying the chance constraint model P5 based on the proposed genetic algorithms, to the same given real-world problem. This is to be based on the experimental conditions described and results achieved so far, in terms of rebalancing demand data over the period of 74 days, including: 29 areas and two rebalancing scenarios, and the optimal parameter combinations of the eight genetic algorithms given in Table 7. We take the real rebalancing demands as input data and compute the solutions using each of the proposed eight algorithms. The results are shown in Figure 22.

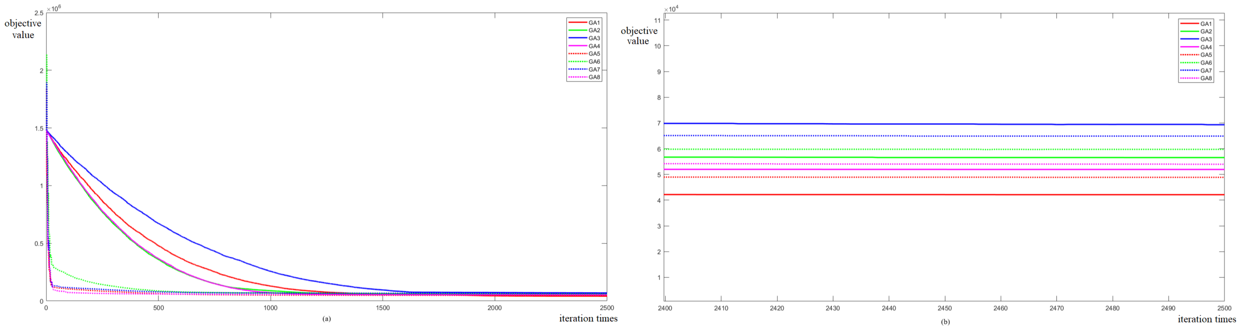

Figure 22: Objective values computed by eight genetic algorithms for a real-world problem case.

The baseline satisfaction level $\gamma$ is set to 80%, reflecting the true level desired by the bike-sharing servicing company. It is clear from the results that the satisfaction levels gained from resolving the proposed model using either of the eight methods are all higher than 80%. This shows the effectiveness of the present approach.

In addition, we compare the effectiveness of the individual evolutionary mechanisms by examining the results of the eight genetic algorithms as provided in Figure 22. From which, we can draw several important conclusive observations:

- Among the proposed eight genetic algorithms, GA1 is able to obtain the most favourable results when each iterates to 2500 generations.
- Genetic algorithms that adopt BCM show better convergence than the others, only requiring iterations to be carried out till the 500th generation while the calculation is also faster, which is ideal for companies that must make rapid rebalancing decisions under pressure of time.
- From the optimal solutions obtained by all these algorithms investigated, certain methods that use BCM yield better results than those using PCM, indicating that the combined use of update and mutation mechanisms can enable a genetic algorithm to jump out of local optimal solutions.

## 6. Conclusion

We have studied the challenging problem of imbalanced distribution of bikes in a bike-sharing network from the entire-city perspective. First, by analysing the real operational data of a bike-sharing company in Beijing, we found that the rebalancing demand in each area has stochastic characteristics with multiple demand scenarios. Second, we established an integer programming model to resolve the static rebalancing problem, which takes multiple rebalancing vehicles with time-varying rental costs into consideration. We demonstrated the inherent properties of the proposed approach, via the adoption of a chance constraint programming model, while revealing the inability of deterministic programming models to solve realistic problems. In addition, we designed eight stochastic simulation-based genetic algorithms running on variable mutation operators together with different evolutionary mechanisms to address the real-world bike-sharing rebalancing problem. The results have demonstrated that among the proposed genetic algorithms, those with block crossover mechanism have stronger convergence ability and are more suitable for companies that must take rebalancing operational decisions promptly. We have also shown that the update and mutation mechanisms enable the implemented algorithms to jump out of local optimal solutions.

The underlying general ideas of the proposed approach not only can be applied in the field of rebalancing operations of shared bikes, but also can be adapted to deal with similar problems in other industrial settings where asymmetric demand patterns may appear in the operational areas, such as rebalancing the batteries of shared cars and shared electric bicycles as well as shared power bank or umbrella across a given metropolitan city.

Future studies will focus on the rebalancing operations under dynamic rebalancing scenarios, the location and inventory design of bike sharing system and reinforce the design of the operators in different heuristic algorithms, such as ant colony algorithm, particle swarm algorithm and adaptive large neighborhood search algorithm.

## Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (No. 71722007, 71931001), and partly by a Sêr Cymru II COFUND Fellowship, UK.

## References

[1] E. Fishman, S. Washington, N. Haworth, A. Watson, Factors influencing bike share membership: an analysis of melbourne and brisbane, Transportation Research Part A: Policy and Practice 71 (2015) 17–30.

[2] K. B. Campbell, C. Brakewood, Sharing riders: How bikesharing impacts bus ridership in new york city, Transportation Research Part A: Policy and Practice 100 (2017) 264–282.

[3] E. Fishman, S. Washington, N. Haworth, Bike share's impact on car use: Evidence from the united states, great britain, and australia, Transportation Research Part D: Transport and Environment 31 (2014) 13–20.

[4] A. Negahban, Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring, European Journal of Operational Research 277 (2019) 317–332.

[5] Z. Tian, J. Zhou, M. Wang, Dynamic evolution of demand fluctuation in bike-sharing systems for green travel, Journal of Cleaner Production 231 (2019) 1364–1374.

[6] Q. Jiang, S.-J. Ou, W. Wei, Why shared bikes of free-floating systems were parked out of order? a preliminary study based on factor analysis, Sustainability 11 (2019) 3287.

[7] Z. Yan, H. Ismail, L. Chen, X. Zhao, L. Wang, The application of big data analytics in optimizing logistics: a developmental perspective review, Journal of Data, Information and Management 1 (2019) 33–43.

[8] J. Liu, M. Chen, H. Liu, The role of big data analytics in enabling green supply chain management: a literature review, Journal of Data, Information and Management 2 (2020) 75–83.

[9] G. Laporte, F. Meunier, R. W. Calvo, Shared mobility systems: an updated survey, Annals of Operations Research 271 (2018) 105–126.

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

[11] T. Raviv, O. Kolka, Optimal inventory management of a bike-sharing station, IIE Transactions 45 (2013) 1077–1093.

[12] Q. Chen, T. Sun, A model for the layout of bike stations in public bike-sharing systems, Journal of Advanced Transportation 49 (2015) 884–900.

[13] J. Schuijbroek, R. Hampshire, W.-J. van Hoeve, Inventory rebalancing and vehicle routing in bike sharing systems, European Journal of Operational Research 257 (2017) 992–1004.

[14] M. Lu, K. An, S.-C. Hsu, R. Zhu, Considering user behavior in free-floating bike sharing system design: A data-informed spatial agent-based model, Sustainable Cities and Society 49 (2019) 101567.

[15] Z. Sun, Y. Li, Y. Zuo, Optimizing the location of virtual stations in free-floating bike-sharing systems with the user demand during morning and evening rush hours, Journal of Advanced Transportation 2019 (2019) 1–11.

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

[17] L. Li, Y. Liu, Y. Song, Factors affecting bike-sharing behaviour in beijing: price, traffic congestion, and supply chain, Annals of Operations Research (2019).

[18] J. Zhang, M. Meng, Z. W. David, A dynamic pricing scheme with negative prices in dockless bike sharing systems, Transportation Research Part B: Methodological 127 (2019) 201–224.

[19] R. Regue, W. Recker, Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem, Transportation Research Part E: Logistics and Transportation Review 72 (2014) 192–209.

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

[21] L. Caggiani, R. Camporeale, M. Ottomanelli, W. Y. Szeto, A modeling framework for the dynamic management of free-floating bike-sharing systems, Transportation Research Part C: Emerging Technologies 87 (2018) 159–182.

[22] J. Brinkmann, M. W. Ulmer, D. C. Mattfeld, Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems, Computers & Operations Research 106 (2019) 260–279.

[23] T. Raviv, M. Tzur, I. A. Forma, Static repositioning in a bike-sharing system: models and solution approaches, EURO Journal on Transportation and Logistics 2 (2013) 187–229.

[24] S. C. Ho, W. Szeto, Solving a static repositioning problem in bike-sharing systems using iterated tabu search, Transportation Research Part E: Logistics and Transportation Review 69 (2014) 180–198.

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

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

[27] F. Cruz, A. Subramanian, B. P. Bruck, M. Iori, A heuristic algorithm for a single vehicle static bike sharing rebalancing problem, Computers & Operations Research 79 (2017) 19–33.

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

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

[30] T. Bulhões, A. Subramanian, G. Erdoğan, G. Laporte, The static bike relocation problem with multiple vehicles and visits, European Journal

of Operational Research 264 (2018) 508–523.

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

[32] Q. Tang, Z. Fu, M. Qiu, A bilevel programming model and algorithm for the static bike repositioning problem, Journal of Advanced Transportation 2019 (2019) 1–19.

[33] B. Lahoorpoor, H. Faroqi, A. Sadeghi-Niaraki, S.-M. Choi, Spatial cluster-based model for static rebalancing bike sharing problem, Sustainability 11 (2019) 3205.

[34] C. A. Poojari, B. Varghese, Genetic algorithm based technique for solving chance constrained problems, European Journal of Operational Research 185 (2008) 1128–1154.

[35] C.-K. Ting, X.-L. Liao, Y.-H. Huang, R.-T. Liaw, Multi-vehicle selective pickup and delivery using metaheuristic algorithms, Information Sciences 406-407 (2017) 146–169.

[36] J. Long, Z. Sun, P. M. Pardalos, Y. Hong, S. Zhang, C. Li, A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem, Information Sciences 478 (2019) 40–61.

[37] H. Zhang, F. Liu, Y. Zhou, Z. Zhang, A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem, Information Sciences 539 (2020) 347–374.

[38] A. Udhayakumar, V. Charles, M. Kumar, Stochastic simulation based genetic algorithm for chance constrained data envelopment analysis problems, Omega 39 (2011) 387–397.