IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 26, 2021, accepted February 4, 2021, date of publication February 8, 2021, date of current version February 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058013

Dynamic Rebalancing Optimization for Bike-Sharing System Using Priority-Based MOEA/D Algorithm

RUNQIU HU[1], ZHIZHENG ZHANG[1], XINWEI MA[2], AND YUCHUAN JIN[3]

[1]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[2]School of Civil and Transportation Engineering, Hebei University of Technology, Tianjin 300401, China
[3]Royal Institute of Technology (KTH), 11428 Stockholm, Sweden

Corresponding author: Xinwei Ma (230169206@seu.edu.cn)

This work was supported in part by the Pre-research Key Laboratory Fund for Equipment under Grant 6142101190304, and in part by the National Key Research and Development Plan of China under Grant 2017YFB1002801.

ABSTRACT As an indispensable part of public transportation systems, the bike-sharing system (BSS) can improve road resource utilization and alleviate traffic congestion, significantly improving urban mobility. The disproportion between the demand and supply creates a giant gap for maintaining the smooth functioning of the system. To address the issue, this paper proposes a dynamic optimization rebalancing model for docked bike-sharing systems, which aims at minimizing the operation cost of rebalancing while maximizing the user satisfaction. The rebalancing demand is evaluated using both historical and predicted data so that a second time service for each station could be avoided within a rebalancing horizon. A time-window based satisfaction modeling is put forward to evaluate user satisfaction. Multiobjective evolutionary algorithm based on decomposition (MOEA/D) under the rolling horizon strategy is adopted to solve the model. To improve the algorithmic performance, local search based on station priority is applied. Numerical experiments using real-world data were implemented to demonstrate the proposed model and the advantage of the improved algorithm. As the results indicate, the proposed algorithm outperforms the nondominated sorting genetic algorithm II (NSGA-II) and MOEA/D without priority-based local search. Solutions with higher satisfaction profit can be discovered with the help of a local search heuristic based on station priority. The experiment also revealed that a slight increase of 7.02% (¥29.2) in the rebalancing cost could yield a significant growth of 1233.7% (¥288.7) in satisfaction profit.

INDEX TERMS Bike-sharing, dynamic rebalancing, multiobjective optimization.

I. INTRODUCTION

Over the past decades, bike-sharing systems (BSSs) have been developed into a flexible, healthy and environmentally friendly transportation mode, and have expanded rapidly all over the world [1]. Currently, more than 2000 BSSs are operating in the world and more than 300 systems are in planning phase or under construction [2]. The current BSSs around the world can be classified into two categories: docked BSSs and dockless BSSs [3]. In the docked BSS, users rent bikes from designated docking stations and then return them to available lockers in the docking stations. The dockless BSSs has two prevalent practices for parking bikes. The first one is using physical or geo-fencing designated parking areas provided in public space with or without bike racks. In the second practice, bikes could be scattered almost everywhere as long as it is at a location accessible to all users [4]. Both types of the BSSs have improved urban mobility. However, the asymmetric spatial and temporal distribution of user demand lead to both systems' imbalance [5]. Therefore, BSS rebalancing is operated to relocate bikes to achieve a desired distribution.

Various approaches for rebalancing bikes can be divided into user-based rebalancing and operator-based rebalancing [6]. The user-based rebalancing approaches usually incentivize the users in the bike rebalancing process, encouraging them to pick or return bikes in specific stations in exchange for monetary incentives [6]. However, the effectiveness of station rebalancing is affected by the willingness

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

of user participation and incentives may become specific budget constraints. Therefore, the operator-based rebalancing approaches are adopted more widely than the former one in many BSSs [7]. In the process of operator-based BSS rebalancing, vehicles are utilized starting from the depot with initial loads of bikes, to pick up or drop off certain number of bikes at surplus or deficit stations, and finally returning to their original depot [8]. Specially, operator-based bike rebalancing optimization can be classified into static optimization and dynamic optimization [9]. For the static problem, the rebalancing operation is performed during the night assuming that the demand for bikes is unchanged, and the aim is to arrange bikes in the system for the next working day. The dynamic one is operated during the day and the routes need to be updated regularly to handle the varying demand. When the system is in use, it is very intricate to manage as it incorporates a scheduling component derived from the users' activity into the operation [10]. Recently, certain amount of studies on dynamic BSS rebalancing have been proposed. Notably, Brinkmann *et al.* considered a stochastic dynamic inventory routing problem (SDIRP) in BSSs [11]. They modeled the problem as a Markov decision process and a dynamic lookahead policy is applied for future demands prediction, minimizing the unsatisfied demands. Schemes for rebalancing are given but costs of the rebalancing are neglected when making the repositioning plan. Caggiani *et al.* proposed a dynamic rebalancing model which aims at achieving a high-level user satisfaction while minimizing the rebalancing costs simultaneously, and a fuzzy decision support system was designed [12]. The work was later expanded to free-floating systems in [13]. Kloimüllner *et al.* developed a dynamic rebalancing method based on their previous static rebalancing method, which minimized the unfulfilled demand as well as the driving time and the number of loading instructions [14]. A framework is proposed in [15] by O'Mahony *et al.*, where a clustering optimization was used to deal with rush hour usage, ensuring that users are always close to available bikes or docks. Chiariotti *et al.* introduced a BSS rebalancing model, which consisted of both user incentives and traditional truck-based rebalancing to minimize the time of a station being full or empty. They formulated the potential departures and arrivals as Poisson processes, using historical arrival/departure rates as means of the probabilistic distribution [16]. However, none of the aforementioned studies considered the station priority when dealing with the dynamic rebalancing problem. As pointed out in [17], in operator-based rebalancing, the ability of the operator to redistribute bikes is limited and priority choices have to be made. Therefore, to enhance the operation and management and improve user satisfaction, a model considering priority of BSS stations is necessary. Specifically, the dynamic optimization rebalancing model proposed in this paper aims at minimizing the operation cost of rebalancing while maximizing the user satisfaction during the BSS rebalancing process. Particularly, the main contributions of this paper lie in:

1) A multiobjective dynamic rebalancing model which simultaneously optimizes of user satisfaction and rebalancing cost is formulated, where the satisfaction is modeled based on time-window and refined to take both timeliness and rebalancing amount into account.
2) Priority of stations is evaluated using a multi-ranking attribute approach and a novel priority-based heuristic is introduced in the MOEA/D algorithm to solve the proposed optimization model.

The remainder of the paper is organized as follows. A literature review of BSS rebalancing optimization is provided in the next section. Subsequently, the paper introduces the problem description and formulation. Afterwards, the proposed method and numerical studies are presented. The conclusions and suggestions for future research are summarized in the last part of the paper.

II. LITERATURE REVIEW

In this section, we will first introduce the objectives of the BSS rebalancing optimization models in the existing studies. Then we will review how current research addresses the determination of rebalancing demand. Next, we will discuss the existing works on modelling of user satisfaction in BSS rebalancing. Afterwards, we will report on the algorithms solving the optimization models. Finally, we will illustrate the research gap between our study and the existing literatures.

A. OBJECTIVES OF BSS REBALANCING

As an extension of the vehicle routing problem, minimizing the cost of BSS rebalancing is the most common objective in the existing research. Lee *et al.* minimized the bike relocating time composed of travel time and time of bike loading and unloading [18]. Zhu *et al.* minimized the total distance travelled between stations and the depots during the repositioning process [19]. Dell' Amico *et al.* minimized the traveling costs using a destroy and repair algorithm to solve the problem [20]. Apart from these, the satisfaction of customers has also been considered in previous works. A model minimizing total unmet demand was proposed by Contardo *et al.* [21]. Brinkmann *et al.* identified an optimal policy which leads to the minimum of expected penalties, i.e., due date violations [22]. Jia *et al.* brought up an optimization model minimizing the total rebalancing time and the number of shortage events during the concerned period [23].

Besides single-objective optimization, some literatures proposed models with multiple objectives. Caggiani *et al.* minimized relocation and lost user costs (i.e., costs for users whose service request cannot be satisfied) at the same time [24]. Dell' Amico *et al.* aimed at minimizing the traveling costs as well as the penalty for unfulfilled demand [2]. You *et al.* proposed a model that minimizes the total cost of failing to meet bike rental and return requests and the total traveling cost and solve the bike repositioning problem with a two-phase heuristic approach [25]. Regue *et al.* modeled a problem considering maximizing the utility gained by visiting a station with a large inefficiency and its neighborhood that is expected to be in deficit, meanwhile minimizing the travel time involved in going to those stations [26]. In recent years, some researchers also took environmental influence in BSS rebalancing into account, notably, total fuel and CO_2 emission costs, as well as unmet demand were minimized simultaneously by Shui *et al.* for dynamic rebalancing [27].

B. DETERMINATION OF REBALANCING DEMAND

Regarding the determination of rebalancing demand, some stochastic approaches were adopted. Dell' Amico *et al.* proposed that demands at each station were represented by random variables, with associated probability distributions, which depend on stochastic scenarios [2]. Kloim'ullner *et al.* modeled the rebalancing demand as an essentially arbitrary function [14]. Besides, a series of studies established optimization models to determine the best rebalancing demand with different objectives. Di *et al.* decided the rebalancing amount from a user demand model [28]. Kadri *et al.* proposed a model for the best inventory of the stations in the system to fulfil at best the demand while minimizing the undesirable situations [29]. Zhu *et al.* calculated the demand in each region by giving corresponding proportional coefficients in different radius ranges through a concentric circle model [19]. Jia *et al.* set the planned parked bikes at each station as a decision variable in the rebalancing optimization model aiming at minimizing the total rebalancing time [23]. In addition, a few works also utilized forecast data to determine the rebalancing demand. Lee *et al.* determined the number of relocation bikes by comparing the predicted demand with current inventory [18]. Regue *et al.* used historical data to predict the inventory of each station by applying the queuing theory model and formulated an optimization model ensuring the number of bikes remains inside the range of the maximum and minimum in historical records in the next period after rebalancing [26]. Chiariotti *et al.* used Poisson processes to formulate the potential departures and arrivals, which uses historical arrival/departure rates as means of the probabilistic distribution [16].

C. SATISFACTION MODELING

As is pointed out above, user satisfaction plays a crucial part in dynamic rebalancing. Lee *et al.* presented a measurement called *Bike Demand Satisfaction Ratio*, which is the percentage of the number of stations that have sufficient inventory to meet bike demand [18]. Rainer-Harbach *et al.* [30] and Kloimüllner *et al.* [14] modeled the dissatisfaction of users by the deviation of the actual number of bikes and the predetermined balanced number of bikes. Some penalty-based approaches were also adopted. Kadri *et al.* used a weighted objective function minimizing the sum of weighted times, where the weight is the gap between current number of bikes a confidence level at each station [31]. Caggiani *et al.* established a fuzzy logic system to measure the satisfaction of users through the rental and return status of a station and then determined the starting time of rebalancing accordingly [24]. The user dissatisfaction was defined by Raviv *et al.* as the expected number of commuter requests (both rentals and returns) that will be rejected in a future time [32] and was later integrated in a novel non-linear time-space network flow model by Zhang *et al.* [33].

D. SOLVING ALGORITHMS

Generally, algorithms solving the rebalancing problems of BSS can be divided into two categories, namely exact methods and inexact methods. For exact algorithms, Contardo *et al.* used *column generation* combined with *benders decomposition* to solve a dynamic rebalancing problem for BSS [21]. Caggiani *et al.* solved the problem minimizing cost and lost users by *branch-and-bound algorithm* [24]. However, due to the complexity of the rebalancing problem, it is intractable to use exact methods to solve large, realistic repositioning problems [34]. Therefore, heuristic algorithms have become prevalent approaches in solving such problems for solutions given limited time [23]. Noticeably, Dell' Amico *et al.* handled the problem with heuristic algorithms based on correlations [35] and later developed a destroy and repair algorithm [20]. You *et al.* adopted a heuristic approach based on linear programming [25]. For metaheuristic algorithms, Lee *et al.* and Zhu *et al.* solved the given problem with genetic algorithms [18], [19]. Jia *et al.* proposed a *Modified Artificial Bee Colony* (*MABC*) in order to find the optimal routes and desired parked bikes of stations [23]. Meanwhile, hybrid algorithms have also been put forward, such as ant-colony optimization and Constraint Programming (CP) [36] as well as Large Neighborhood Search (LNS) and CP [28] by Di Gaspero *et al.* It is worth mentioning that for dynamic cases, a series of work applied a rolling horizon approach to decrease the scale of the problem, e.g., [27] by Shui *et al.*, [33] by Zhang *et al.* and [37] by Mellou *et al.*

E. RESEARCH GAP

Multiple existing studies attempt to improve the user satisfaction quantitatively, either by minimizing the total unmet demand [2], [21] or deviation from target fill level of a station [38]. At the same time, some choose to cope with satisfaction temporally by minimizing the length of time for a station short/surplus of bikes [31]. However, none of the methods above dealt with timeliness and quantity simultaneously. In our study, we refine the satisfaction modeling based on a time-window allowing the tardiness of service, which gives consideration to both timeliness and demand satisfaction. Furthermore, to the best of our knowledge, none of the literatures reviewed above considered and utilized the station priority in the solving algorithms. Therefore, in this paper, the rebalancing model proposed is solved via a priority-based MOEA/D (PB-MOEA/D) algorithm. The algorithm decomposes the proposed multiobjective problem into a set of scalar subproblems and a priority-based heuristic is introduced to improve the solution and the algorithmic performance.

III. PROBLEM DESCRIPTION AND FORMULATION

A. PROBLEM DESCRIPTION

In this paper, we focus on a BSS composed of a certain number of bike stations within a specific area, which is a 2-D space $X \subseteq \mathbb{R}^2$. Within this space, $S := \{S_1, S_2, \ldots, S_N\} \subseteq X$ is defined as the location set of bike stations which are used to deliver and pick up public bikes. $s_c \in X$ is defined as the depot, which is used to store bikes and rebalancing trucks. The dynamic rebalancing of BSS in this paper is the process in which the truck departs from the depot with some initial load of bikes, and the rebalancing plan is made and updated according to the constant change on demand and inventory of each station over time.

During rush hour in the morning, many users have commuting needs, and the demand for renting and returning bikes sharply increases. Therefore, we choose rush hour in the morning, e.g., 8:00 A.M. to 9:00 A.M. as our study horizon, denoted as $[0, \tau]$, which is divided into 4 periods with equal length ζ and $\tau = 4\zeta$. The rebalancing strategy is determined and updated at the beginning of each stage based on current demand and inventory of each station. In the first stage, an initial vehicle rebalancing plan concerning the whole horizon that meets the constraints is determined based on the initial information of stations and trucks. After the execution of the plan for a period of time ζ, the next stage arrives and the rebalancing strategy concerning the remaining horizon is updated according to the dynamic changes of the demand and inventory. The process is repeated until the end of the horizon as shown in Figure 1.

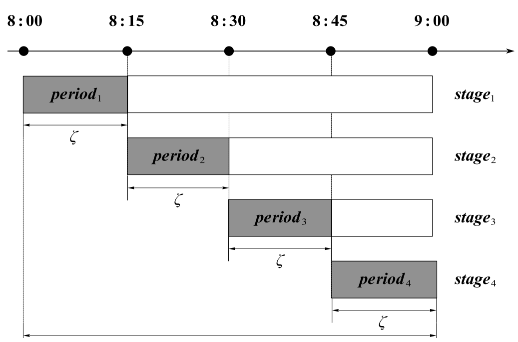


FIGURE 1. Description of the rebalancing horizon.

The trucks set off from the depot and travel to all the stations with demand to pickup surplus bikes or deliver insufficient bikes, and the rebalancing demand for each station is calculated in order to avoid a second-time rebalancing during the horizon, therefore each station that requires rebalancing can be served at most one time. At the end of rebalancing, each truck should return to the depot. The key problem is deciding a route for each truck inside the horizon that can minimize the rebalancing cost and maximize the user satisfaction simultaneously.

B. MATHEMATIC FORMULATION

The dynamic rebalancing problem can be formulated as follows. First, we list all the symbols and mathematical notations used in the article, which contains set, parameters and the decision variables. Then we give the mathematic model based on these notations, including the objectives and the constraints.

1) SETS

- S the set of all stations inside the current service window
- S_0 the set of all stations and the depot, $S_0 = S \cup \{s_c\}$ where s_c is the depot.
- K the set of all trucks

2) PARAMETERS

- t_0 initial time of the rolling period (unit: minute)
- v velocity of the truck (unit: m/min)
- U capacity of the truck (unit: bike)
- $dis_{i,j}$ distance between station i and j (unit: m)
- ω_r cost of single run (unit: ¥/truck)
- ω_{st} loading and unloading cost (unit: ¥/min)
- ω_d travelling cost for one car (unit: ¥/min)
- $demand_i$ the rebalancing demand. $demand_i > 0$, $demand_i$ bikes need to be delivered to station i; $demand_i < 0$, $-demand_i$ bikes need to be picked up from station i (unit: bike)
- C_i the number of docks at station i
- g satisfaction profit for one bike at its maximum satisfaction (unit: ¥)
- $init_i$ initial number of bikes at station i in the rolling period (unit: bike)
- r_i variation rate of the bike number at station i (unit: bike/min)
- $DMAX$ travelling distance limit for one truck (unit: km)
- $prior_i$ priority of station i

3) DECISION VARIABLES

- $x_{i,j,k}$ a binary variable that defines the route of vehicle k, if $x_{i,j,k} = 1$, truck k travel from station i to station j, 0 otherwise
- y_k a binary variable that defines the use of truck, if $y_k = 1$, truck k is used in this rebalancing operation, 0 otherwise
- $l_{i,k}$ actual rebalancing amount of truck k at station i, if $l_{i,k} > 0$, truck k delivers $l_{i,k}$ bikes at station i, if $l_{i,k} < 0$, truck k picks up $-l_{i,k}$ bikes at station i
- $q^+_{i,k}$ the number of bikes on truck k when it leaves station k
- $q^-_{i,k}$ the number of bikes on truck k when it arrives at station k

$m_{i,k}^{+}$	the number of bikes at station i when truck k arrives
$m_{i,k}^{-}$	the number of bikes at station i when truck k leaves
$t_{i,k}^{+}$	the time when truck k leaves station k
$t_{i,k}^{-}$	the time when truck k arrives at station k
w_i	satisfaction profit of station i
$twork_{i,k}$	working time of truck k at station i

4) MATHEMATIC MODEL

The proposed model consists of two objectives. The first objective (1) of the route optimization model is minimizing the rebalancing costs, which consists of the distance cost, fixed running cost for each truck and the working time cost at each station.

$$\min F_1 = \omega_d \sum_{i\in S_0}\sum_{\substack{j\in S_0\\ j\neq i}} x_{i,j,k}\cdot\frac{dis_{i,j}}{v} + \omega_r\sum_{k\in K} y_k + \omega_{st}\sum_{i\in S_0}\sum_{\substack{j\in S_0\\ j\neq i}}\sum_{k\in K} x_{i,j,k}\cdot twork_{j,k} \quad (1)$$

The second objective (2) of the route optimization model aims at maximizing the overall satisfaction profit, which will be discussed in detail in section IV-A. Here g is the satisfaction profit for one bike at its maximum satisfaction, and in this paper we set g as ¥1.

$$\max F_2 = g\sum_{i\in S_0}\sum_{\substack{j\in S_0\\ j\neq i}}\sum_{k\in K} x_{i,j,k}\cdot w_i \quad (2)$$

Several constraints have to be satisfied, and will be introduced in the rest of this part. First, the total rebalancing time of trucks cannot exceed the length of the rolling period, as is shown in constraint (3).

$$\sum_{i\in S_0}\sum_{\substack{j\in S_0\\ j\neq i}} x_{i,j,k}\left(\frac{dis_{i,j}}{v} + twork_{j,k}\right) \le \zeta, \quad \forall k\in K \quad (3)$$

When picking up or delivering bikes at each station, the number of bikes picked up from or delivered to a station should not only meet the rebalancing demand of the station, but also follow the constraints of the truck capacity and the available bikes at the station, as is shown in constraint (4).

$$l_{i,k} = \begin{cases} \min\left\{demand_i, q_{i,k}^{+}, C_i - m_{i,k}^{+}\right\}, & demand_i > 0,\\ -\min\left\{-demand_i, U - q_{i,k}^{+}, m_{i,k}^{+}\right\}, & demand_i < 0,\end{cases} \quad \forall i\in S,\quad \forall k\in K \quad (4)$$

After rebalancing, the number of bikes of both stations and trucks are updated, which will be calculated through constraint (5)-(8). Specifically, constraint (5) represents the update of the number of bikes on the truck when it leaves a station. Constraint (6) represents the number of bikes on the truck when it arrives at a station. Constraint (7) represents the number of bikes available at a station when the truck arrives at the station. Constraint (8) represents the number of bikes available at a station when the truck leaves the station.

$$q_{i,k}^{-} = q_{i,k}^{+} - l_{i,k}, \forall i\in S, \forall k\in K \quad (5)$$

$$q_{i,k}^{+} = \sum_{\substack{j\in S_0\\ j\neq i}} x_{j,i,k}\cdot q_{j,k}^{-}, \forall i\in S, \forall k\in K \quad (6)$$

$$m_{i,k}^{+} = init_i, \forall i\in S, \forall k\in K \quad (7)$$

$$m_{i,k}^{-} = init_i - \sum_{j\in S}\sum_{k\in K} x_{j,i,k}\cdot l_{i,k}, \quad \forall i\in S, \forall k\in K \quad (8)$$

Besides the inventory update, the following constraints keep track of the time when the truck arrives or leaves at a station. Constraint (9) indicates that the time when the truck arrives at a station is the time when the truck leaves the previous station plus the time if travelling from the previous station to this station. Constraint (10) means the time when the truck leaves a station is the time when the truck arrives at the station plus the time for loading and unloading of the bikes at this station.

$$t_{i,k}^{+} = \sum_{\substack{j\in S_0\\ j\neq i}} x_{j,i,k}\cdot t_{j,k}^{-} + \sum_{\substack{j\in S_0\\ j\neq i}} x_{j,i,k}\cdot\frac{dis_{j,i}}{v}, \quad \forall i\in S, \forall k\in K \quad (9)$$

$$t_{i,k}^{-} = t_{i,k}^{+} + twork_{i,k}, \quad \forall i\in S, \forall k\in K \quad (10)$$

Constraints (11) and (12) limit that in a single rebalancing process, each station in the concerning service window will be served and only served once.

$$\sum_{k\in K}\sum_{\substack{j\in S_0\\ j\neq i}} x_{i,j,k} = 1, \quad \forall i\in S \quad (11)$$

$$\sum_{k\in K}\sum_{\substack{j\in S_0\\ j\neq i}} x_{j,i,k} = 1, \quad \forall i\in S \quad (12)$$

Constraint (13) represents that $x_{i,j,k}$ is a 0-1 binary variable.

$$x_{i,j,k}\in\{0,1\}, \forall i,j\in S_0, i\neq j, \quad \forall k\in K \quad (13)$$

Constraint (14) means that once the truck conducts the rebalancing, it has to depart from the depot.

$$y_k = \sum_{j\in S} x_{s_c,j,k}, \quad \forall k\in K \quad (14)$$

Constraint (15) means that the truck travel distance cannot exceed the maximum driving distance of a truck.

$$\sum_{i\in S_0}\sum_{\substack{j\in S_0\\ j\neq i}} x_{i,j,k}\cdot dis_{i,j} \le DMAX, \quad \forall k\in K \quad (15)$$

IV. PROPOSED METHOD

A. DEMAND EVALUATION

The determination of rebalancing demand is critical to solving dynamic rebalancing problems. After the rebalancing demand is determined, the dynamic rebalancing problem can be transformed into a pickup-delivery problem with capacity and time window constraints. The number of bikes at the station after service is supposed to maintain within a reasonable range before the end of the period to avoid the same station being visited by trucks for a second time, which might bring unnecessary cost. Therefore, a method of evaluating the rebalancing demand is proposed, which utilizes the historical and predicted rental and return data together and aims at avoiding a second time rebalancing for each station within the rebalancing horizon.

Let the initial time of the rolling horizon be t_0 and the horizon length be τ. Each rolling horizon is equally spaced into $n = \lceil \tau/\zeta \rceil$ rolling periods of length ζ, i.e., $[t_0, t_0 + \zeta)$, $[t_0+\zeta, t_0+2\zeta)$, ..., $[t_0+(n-1)\zeta, t_0+n\zeta]$. The number of initial bikes station i in the m^{th} rolling period is $init_{i,m}$, then the available bike at any time $t \in [t_0 + (m-1)\zeta, t_0 + \tau]$, denoted as $avail_{i,m}(t)$ would be (16)

$$avail_{i,m}(t) = init_{i,m} - r_{i,m} \cdot [t - t_0 - (m-1)\zeta]$$
$$(t_0 + (m-1)\zeta \le t \le t_0 + \tau, m = 1, \ldots, n) \quad (16)$$

where $r_{i,m} = \lambda r^{past}_{i,m,\zeta} + (1-\lambda)\, r^{pred}_{i,m,\tau}$ $(0 \le \lambda \le 1)$ is the variation rate of the bike number at station i, and is determined by the past variation rate $r^{past}_{i,m,\zeta}$ within time interval $[t_0+(m-2)\zeta, t_0+(m-1)l]$ and the predicted variation rate $r^{pred}_{i,m,\tau}$ within time interval $[t_0+(m-1)\zeta, t_0+\tau]$. λ is a weight coefficient and is set as 0.5 in this paper. These two rates are calculated through the past and predicted rental and return bike numbers respectively (denoted as N with corresponding superscript and subscript)

$$r^{past}_{i,m,\zeta} = \frac{N^{past,rent}_{i,m,\zeta} - N^{past,return}_{i,m,\zeta}}{\zeta} \quad (17)$$

$$r^{pred}_{i,m,\tau} = \frac{N^{pred,rent}_{i,m,\tau} - N^{pred,return}_{i,m,\tau}}{\tau - (m-1)\zeta} \quad (18)$$

We intend to make the number of available bikes at every station inside a safe range ($[0.2C_i, 0.8C_i]$) through the interference of rebalancing [39]. Therefore, we investigate the initial number of bikes $init_{i,m}$ and available bikes at the end of the rolling horizon $avail_{i,m}(t_0+\tau)$ and compare them with the safe range of station i to determine the rebalancing demand. If the number of bikes of a station at the end of the horizon is outside the safe range, we call the station an *imbalanced station*. Figure 2(a)-(c) shows bike number variation curves regarding time of different imbalanced stations. Specifically, Figure 2(a) shows a station with delivery demand, where the number of bikes at the station would drop below the lower bound ($0.2C_i$) at $t^{warning}_{i,m}$ with the current variation rate, thus when the rebalancing truck arrives at station i at $ts_{i,m}$, the rebalancing could be viewed as an upper shift of the curve, aiming at ensuring the number of available bikes at this station over $0.2C_i$ before the horizon ends. Similarly, Figure 2(b) shows the station with pickup demand whose

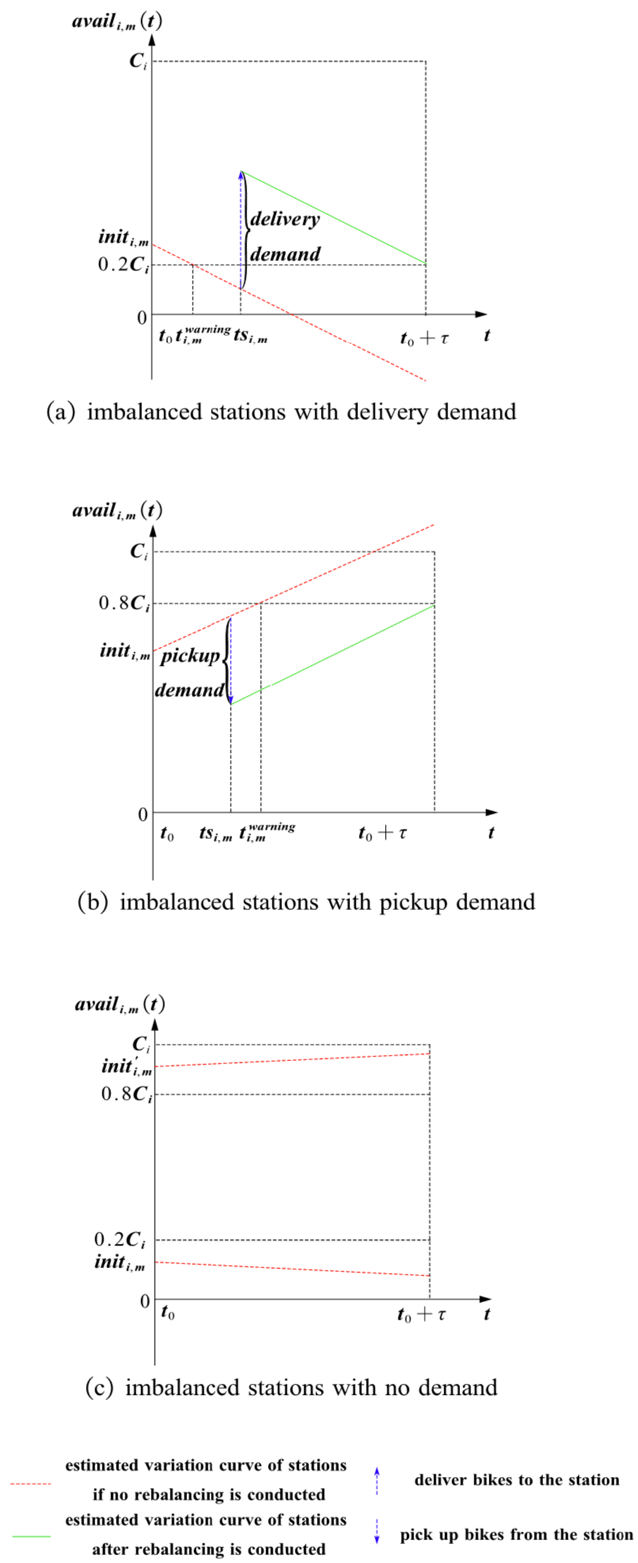


FIGURE 2. Bike number variation curves of imbalanced stations.

inventory would exceed the upper bound ($0.8C_i$) at $t_{i,m}^{warning}$, thus the rebalancing could be viewed as a downer shift of the curve, aiming at ensuring the number of available bikes at this station below $0.8C_i$ before the horizon ends. For most of the stations, we can conclude that if a station is an imbalanced station, the station needs rebalancing and the demand would be the deviation of the current inventory and its closest bound. However, there are two exceptions shown in Figure 2(c). The two curves illustrate two conditions where the inventory at the end of the horizon is out of the safe range but no rebalancing demand is required. Obviously, the inventory of these two kinds of stations are changing in a relatively small rate. Therefore, these stations are not likely to be inconvenient for renting or returning bikes despite its bike number at end of the horizon being outside the safe range. Therefore, rebalancing for these stations is of little utility. So if the number of bikes of a station at the end of the horizon is over upper bound (below lower bound) of the safe range, but its initial inventory is less than C_i (more than 0), we treat the station as one with no rebalancing demand. Table 1 discusses all the possible scenarios and gives the rebalancing target accordingly. $\uparrow 0.2C_i$ means increasing the number of bikes in the station to $0.2C_i$, $\downarrow 0.8C_i$ means decreasing the number of bikes in the station to $0.8C_i$ and / means no rebalancing is required.

TABLE 1. Different Scenarios of Rebalancing Demand.

$avail_{i,m}(t_0+\tau)$ \ $init_{i,m}$	$[0, 0.2C_i)$	$[0.2C_i, 0.8C_i)$	$[0.8C_i, C_i]$
≤ 0	$\uparrow 0.2C_i$	$\uparrow 0.2C_i$	$\uparrow 0.2C_i$
$(0, 0.2C_i)$	/	$\uparrow 0.2C_i$	$\uparrow 0.2C_i$
$[0.2C_i, 0.8C_i)$	/	/	/
$[0.8C_i, C_i)$	$\downarrow 0.8C_i$	$\downarrow 0.8C_i$	/
$> C_i$	$\downarrow 0.8C_i$	$\downarrow 0.8C_i$	$\downarrow 0.8C_i$

Based on Table 1, the rebalancing demand of each station i at rolling period m can be calculated through (19)

$$demand_{i,m} = \begin{cases} avail_{i,m}(t_0+\tau) - 0.8C_i, & \begin{array}{c} avail_{i,m}(t_0+\tau) > C_i \text{ or} \\ 0 < init_{i,m} < 0.8C_i < \\ avail_{i,m}(t_0+\tau) < C_i \end{array} \\ 0.2C_i - avail_{i,m}(t_0+\tau), & \begin{array}{c} avail_{i,m}(t_0+\tau) \leq 0 \text{ or} \\ 0 < avail_{i,m}(t_0+\tau) < \\ 0.2C_i < init_{i,m} \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

B. STATION PRIORITY EVALUATION

Dynamic rebalancing of BSS is a limited resource allocation problem, and due to the inherent feature of the problem, stations with higher priority should be considered first. To evaluate the importance of a bike station and its location impact in the bike share network, we propose a multi-ranking attribute method based on TOPSIS [40], which uses rebalancing demand (denoted as *dem*), the time before the inventory running outside the safe range (denoted as *tw*), and the minimum distance from stations to trucks (denoted as *dist*) as three criteria for station priority evaluation. The process goes as follows:

1) **Step 1: Quantification of criteria.** Among the three criteria, *dem* has positive impact on the priority, as more demand indicates more priority, while *tw* and *dist* has negative impact has negative impact on the priority, as more time the inventory running outside the safe range and more distance from stations to trucks indicate that the station has less priority. Therefore, the quantified evaluation matrix $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{n\times 3}$ (20) is made up of 3 columns, where the elements in the first column are values of corresponding demand of each station i (i.e., $x_{i1} = dem_i$) and the rest two are the values of tw_i and $dist_i$ after a transformation by subtraction with 1 due to their negative impact on priority (i.e., $x_{i2} = 1 - tw_i$, $x_{i3} = 1 - dist_i$). Here $i = 1, 2, \ldots, n$ and n is the number of stations.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} \end{bmatrix} \tag{20}$$

2) **Step 2: Normalization of the evaluation matrix.** The normalized evaluation matrix $\mathbf{N} = (n_{ij}) \in \mathbb{R}^{n\times 3}$ is calculated through (21), where x_{ij} is the element from (20).

$$n_{ij} = \frac{x_{ij}}{\sum_{i=0}^{n} x_{ij}}, \quad \forall i \in \{1, 2, \ldots, n\}, \forall j \in \{1, 2, 3\} \tag{21}$$

3) **Step 3: Construction of the weighted normalized matrix.** The weighted normalized matrix denoted as $\mathbf{Y} = (y_{ij}) \in \mathbb{R}^{n\times 3}$ is calculated through (22), where ω_j is the weight of the j^{th} criterion calculating using the entropy weight method [41].

$$y_{ij} = n_{ij} \cdot \omega_j, \quad \forall i \in \{1, 2, \ldots, n\}, \forall j \in \{1, 2, 3\} \tag{22}$$

4) **Step 4: Calculation of the ideal solution.** Determine the positive ideal solution Y^+ and negative ideal solution Y^- with (23) and (24), where Y is the weighted normalized matrix calculated in step 3.

$$Y^+ = (Y_1^+, Y_2^+, Y_3^+) = (\max_{1\leq i\leq n} y_{i1}, \max_{1\leq i\leq n} y_{i2}, \max_{1\leq i\leq n} y_{i3}) \tag{23}$$

$$Y^- = (Y_1^-, Y_2^-, Y_3^-) = (\min_{1\leq i\leq n} y_{i1}, \min_{1\leq i\leq n} y_{i2}, \min_{1\leq i\leq n} y_{i3}) \tag{24}$$

5) **Step 5: Calculation of the distance to the ideal solution.** According to the positive and negative ideal solution Y^+ and Y^-, calculate the distance of each station to

the ideal solution using (25) and (26).

$$D_i^+ = \sqrt{\sum_{j=1}^{3}(y_{ij} - Y_j^+)^2}, \quad i = 1, 2, \ldots, n \quad (25)$$

$$D_i^- = \sqrt{\sum_{j=1}^{3}(y_{ij} - Y_j^-)^2}, \quad i = 1, 2, \ldots, n \quad (26)$$

6) **Step 6: Calculation of closeness degree.** The closeness degree of station i, denoted as C_i, represents the closeness of a station to the ideal station, and is calculated by (27) the distance D_i^+ and D_i^- obtained in step 5.

$$C_i = \frac{D_i^-}{D_i^- + D_i^+}, i = 1, 2, \ldots, n \quad (27)$$

7) **Step 7: Calculation of the station priority.** The priority of a station i is eventually calculated by (28)

$$prior_i = exp^{\frac{n-rank_i}{n}} \quad (28)$$

where $rank_i$ is the ranking of the closeness degree obtained in step 6, n is the number of the candidate stations [42].

C. TIME WINDOW AND USER SATISFACTION PROFIT

The best service can be obtained if the demand of station i is met before the number of bikes in a station goes empty or reaches the maximum capacity of the station C_i. In this case, user satisfaction becomes the highest. Given a rolling period m, the initial number of bikes in station i is denoted as $init_{i,m}$, and the variation rate of the station at the period is denoted as $r_{i,m}$. The *expected service time*, denoted as tu_i^{sat}, which is the time when the number of bikes in a station goes empty or full, can be calculated by:

$$tu_i^{sat} = \begin{cases} \min\left\{\tau - t_0 - (m-1)\zeta, t_0 + (m-1)\zeta + \dfrac{init_{i,m}}{r_{i,m}}\right\}, \\ \quad r_{i,m} > 0, init_{i,m} \in [0, C_i] \\ \min\left\{\tau - t_0 - (m-1)\zeta, t_0 + (m-1)\zeta + \dfrac{C_i - init_{i,m}}{-r_{i,m}}\right\}, \\ \quad r_{i,m} < 0, init_{i,m} \in [0, C_i] \\ \tau - t_0 - (m-1)\zeta, \\ \quad r_{i,m} = 0, init_{i,m} \in [0, C_i] \\ t_0 + (m-1)\zeta, \\ \quad init_{i,m} \notin [0, C_i] \end{cases} \quad (29)$$

However, in a real-world rebalancing process, the trucks may not be able to arrive at the station to provide service before the expected service time. If it arrives at the station later than this time, the satisfaction of the user may start to

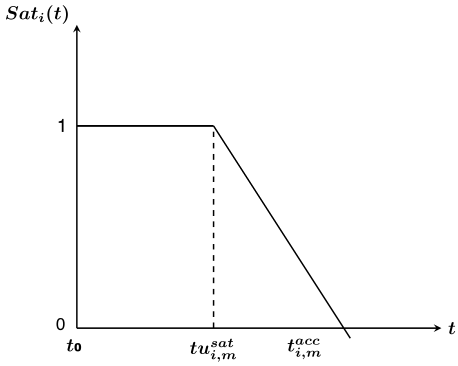


FIGURE 3. One-side Satisfaction Time Window.

decrease, but will not immediately drop to 0. A period of time of tardiness is allowed before the satisfaction decreases to zero. The exact time when the satisfaction becomes 0 is called *accepted time*, denoted as tu_i^{acc}. To make the model closer to the rebalancing problem in practice, we use a one-sided soft time window [43] to model such satisfaction allowing tardiness, as shown in Figure 3. If the service is provided before the expected service time, the user could obtain a satisfaction of 1. Later on, the satisfaction begin to decline and becomes 0 at the accepted time. Therefore, the satisfaction regarding station i regarding time t, denoted as $Sat_i(t)$ can be calculated by

$$Sat_i(t) = \begin{cases} 1, & t_0 < t \le tu_i^{sat} \\ \dfrac{tu_i^{sat} - t}{tu_i^{acc} - tu_i^{sat}}, & tu_i^{sat} < t \le tu_i^{acc} \\ 0, & t > tu_i^{acc} \end{cases} \quad (30)$$

While satisfaction characterizes the timeliness of rebalancing service, the profits obtained from user satisfaction are not only related to the time when the truck arrives at the station, but also the number of demand actually satisfied after the service of the truck. An extreme condition would be that an empty truck arrives at a station with huge demand for bike delivery before tu_i^{sat}, yet no bikes can be delivered. In this case, one could not say the service provides a maximum satisfaction. In addition, for some stations, the demand could be considerably large, thus the impact of loading or unloading time on satisfaction could not be ignored. To this end, we define the user satisfaction profit of station i, denoted as w_i, combining user satisfaction with the actual rebalancing amount. Instead of evaluating the satisfaction only in terms of each station, we made a refinement on the satisfaction to each single bike picked up or delivered to the station. Specifically, for each bike rebalanced, if it is picked up or delivered before the time of the maximum user satisfaction at the station (i.e., before tu_i^{sat} with satisfaction of 1), a profit of g would be achieved. After the expected service time, the satisfaction profits obtained from a bike rebalanced at t will decrease gradually and is evaluated with $g \cdot Sat_i(t)$. Overall, the

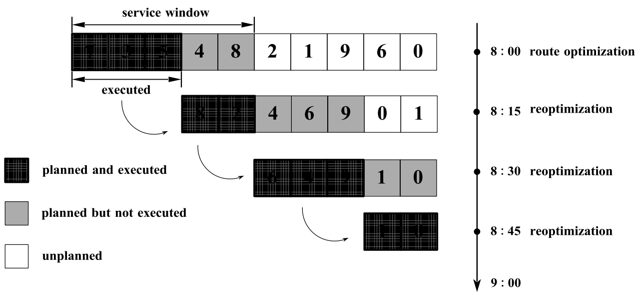


FIGURE 4. Rolling horizon strategy.

satisfaction profit at a station can be formulated as

$$w_i = g \sum_{k \in K} \sum_{\rho=1}^{|l_i|} x_{i,j,k} \cdot Sat_i \left(t_{i,k}^{+} + \rho t_s \right) \tag{31}$$

where $t_{i,k}^{+}$ is the arriving time of truck k at station i, t_s is the time needed for loading or unloading one bike at the station.

D. ROLLING HORIZON STRATEGY

As an extension of the vehicle routing problem (VRP), BSS dynamic rebalancing is an NP-Hard problem [27]. It is difficult to obtain an accurate solution in polynomial time. Using a rolling horizon strategy can divide the problem into multiple subproblems, and the scheme for each period is solved limited inside the current search space by some domain specific criterion. This can not only adapt to the dynamic events, but also reduce the computing cost by dealing with only part of the requests in each time horizon [44].

Figure 4 shows the process of the dynamic rebalancing using rolling strategy. We divide the horizon into 4 stages, each stage having a current *service window*. The window has a fixed limited size (normally smaller than the number of all the stations with service requirement), and stations with higher priority are added to the window until it is full. Optimal route consisting of stations in the *service window* is generated by optimization algorithms and the truck executes the rebalancing according to the scheme until the arrival of next rolling stage. The rebalancing demand, station priority, as well as the stations in the service window are updated dynamically and a new optimal route is calculated. The above procedure loops until the end of the horizon. The complete process of rolling horizon strategy is shown in Algorithm 1.

Algorithm 1 Rolling Horizon Strategy for BSS Rebalancing

Input: S: A set of stations with rebalancing demand, max_stage: the number of stages in a horizon
Output: optimal route list $R_1, R_2, \ldots, R_{max\_stage}$
  $current\_stage \leftarrow 1$,
  $R \leftarrow \emptyset$
  while $current\_stage \leq max\_stage$ **do**
    $|W| \leftarrow$ service window size
    update the demand and priority for each station
    if $|S| \leq |W|$ **then**
      $W \leftarrow S$
    else
      $W \leftarrow$ top $|W|$ stations from S in priority
    end if
    route optimization for stations in W
    $r_{opt} \leftarrow$ select one route from optimal routes by some domain-specific criterion
    $R_{current\_stage} \leftarrow r_{opt}$
    $current\_stage \leftarrow current\_stage + 1$
    $S \leftarrow S \setminus r_{opt}$
  end while
  return $R_1, R_2, \ldots, R_{max\_stage}$

E. PRIORITY-BASED MOEA/D (PB-MOEA/D)

1) MOEA/D

The MOEA/D algorithm adopts the decomposition idea, which was first proposed by Zhang and Li [45]. The algorithm decomposes the problem of approaching the Pareto front into a certain number of single-objective optimization problems, and then uses evolutionary algorithm to solve these single-objective optimization problems simultaneously. The algorithm maintains a population composed of the current optimal solution of each sub-problem. The neighbor relationship between the sub-problems is defined as the distance between the weight vectors of the sub-problems. The optimization process of each sub-problem is through the evolution between its neighboring sub-problems, which successfully introduces the decomposition methods commonly used in mathematical programming into the field of evolutionary multi-objectives optimization [46], and has been proved to be an effective method for solving multiobjective problems (MOP) [45], [47].

In MOEA/D, a neighborhood of weight vector λ_i is defined as a set of its several closest weight vectors in $\lambda_1, \ldots, \lambda_N$. The neighborhood of the i^{th} subproblem consists of all the subproblems with the weight vectors from the neighborhood of λ_i. The population is composed of the best solution found so far for each subproblem. The optimization model proposed in III is a typical MOP with two objectives, which could be solved using MOEA/D. A Pareto-optimal solution is defined as the best solution archived for one objective without disadvantaging at least one of the other objectives [48]. Given an MOP and two different solutions s_1 and s_2, if there is at least one objective value in s_1 superior to corresponding objective in s_2, and the rest objective values in s_1 are not inferior to that of s_2, then s_2 is dominated by s_1. Here we refer to [45] and give the specific algorithm for applying MOEA/D to BSS rebalancing, as shown in Algorithm 2. Note that any mathematical aggregation approach can serve for problem decomposition, and we refer to [49] and use the *Tchebycheff* approach. Moreover, as the algorithm demonstration proposed in the original literature discussed the maximized case, we convert the objective (1) to maximizing its negative.

Algorithm 2 MOEA/D for BSS Rebalancing

Input: MOP: $\max f_1 = -F_1$
$\max f_2 = F_2$,
Stopping Criterion: Algorithm terminates at the max generation G,
N : The number of sub-problems,
$\boldsymbol{W}$: Uniformly distributed weight vectors,
T: Neighborhood number for each vector

Output: R^*: Nondominated Solutions of Routes
EP:External Population

$EP \leftarrow \emptyset$
$R^* \leftarrow \emptyset$
Compute the Euclidean distance between any two vectors in $\boldsymbol{W}$
Generate the initial population of routes $R^1, \ldots, R^N$ by random permutation
$current\_gen \leftarrow 0$
while $current\_gen \neq G$ **do**
  $current\_gen \leftarrow current\_gen + 1$
  for $i \leftarrow 1$ to N **do**
    $B(i) \leftarrow \{i_1, \ldots, i_T\}$ where $i_1, \ldots, i_T$ are the closest T neighbors of λ^i
    $FV^i \leftarrow F(R^i)$, where $F = [f_1, f_2]$
  end for
  Initialize ideal solution $z = (z_1, \ldots, z_m)^T$ by setting the minimum fitness value of each objective among the generated initial population
  for $i \leftarrow 1$ to N **do**
    Randomly pick two indices $s \in B(i), t \in B(i) (s \neq t)$, generate a new route y using genetic operator via R^s and R^t
    Apply route improvement heuristic methods on y including the shortest path and a priority-based heuristic to get a new route y'
    for $j \leftarrow 1$ to 2 **do**
      if $f_j(y') < z_j$ **then**
        $z_j \leftarrow y'$
      end if
    end for
    for $j \in B(i)$ **do**
      if $g^{te}(y' \mid \lambda^j, z) \leq g^{te}(r^j \mid \lambda^j, z)$ **then**
        $r^j \leftarrow y', FV^j \leftarrow F(y')$
      end if
    end for
  end for
  Remove all vectors dominated by $F(y')$ from EP
  Remove all the corresponding routes from R^*
  Add $F(y')$ to EP if no vectors in EP dominate $F(y')$ Add all the corresponding routes to R^*
end while
return EP, R^*

2) PRIORITY-BASED HEURISTIC

It is noticed that a problem-specific improvement of the solution after genetic approach is needed in the update process of MOEA/D. After the new solution y is generated for a given subproblem, an attempt is made to improve it using local search. Three local heuristics to search for better routing solutions in the VRP with time window (VRPTW), namely *Double Shift* (*DS*), *Lambda Interchange* (*LI*), and *Shortest Path* (*SP*) have been used to solve a Tri-Objective Vehicle Routing Problem with MOEA/D, and are associated with different objectives respectively [49]. *DS* is affiliated to the number of vehicles, *LI* is affiliated to the route balancing and *SP* is affiliated to the distance cost. It is observed that *SP* is applicable for one of the objectives in our model, and we adopt it as one of the local heuristics.

At the same time, we notice that the evaluation of the station priority serves as a vital indicator to help decide which station should be served first, thus a priority-based heuristic is designed to improve the quality of a solution. Stations with higher priority will be moved to the front of the routes with higher probabilities based on Roulette Wheel selection method. Specifically, given a sequential encoded route:

$$R = \left[r_{1_1}, r_{1_2}, \ldots, r_{1_{n_1}}, -1, r_{2_1}, r_{2_2}, \ldots, -1, r_{k_1}, \ldots, r_{k_{n_k}}\right] \tag{32}$$

where r_{m_i} represents the i^{th} station visited by the m^{th} truck ($1 \leq m \leq k$) and r_{n_m} is the last station visited by the m^{th} truck, and -1 is used to split the route of each truck. Two problems are of our concern when conducting local search:

i) The distribution of the station priority in the routes assigned to trucks should be relatively balanced, i.e., solutions of some trucks serving mostly prioritized stations and some trucks serving mostly non-prioritized ones are not expected.
ii) Inside the route of each truck, stations with higher priority are supposed to appear at the front of the service queue.

For the first problem, a priority-distribution repairing algorithm is proposed as follows. Let the route of truck i be R_i, for $i = 1, \ldots, K$, where K is the number of trucks:

Step 1 Calculate the range of the priority in each route by

$$rng_i = \max_{j=1,\ldots,n_i} prior_{r_j} - \min_{j=1,\ldots,n_i} prior_{r_j} \tag{33}$$

Step 2 For all the routes of trucks, choose two trucks p, q with the lowest range of priority, which indicates that the priority distributions of these routes are relatively concentrated and are the possible candidate routes in need of improvement.

Step 3 Select the first two stations with the highest priority s_1^h, s_2^h and the last two stations with the lowest priority s_1^l, s_2^l in $R_p \cup R_q$. Then arrange (s_1^h, s_1^l) into the first route and (s_2^h, s_2^l) the second one at a random position. If the new solution dominates the original one, update the original route. Otherwise, the original solution maintains.

Regarding the second issue, a neighboring search using roulette wheel selection is proposed. For each route R_k,

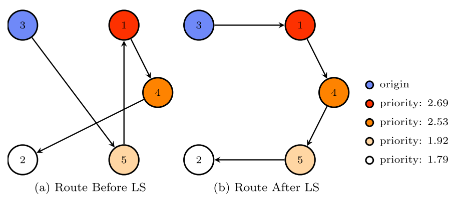


FIGURE 5. **Priority-based Heuristic.**

calculate the selection possibility for each station i in R_k with

$$prob_i = \frac{prior_i}{\sum_{j \in R_k} prior_j} \tag{34}$$

and the accumulated selection probability of each station would be

$$prob_i^{acc} = \sum_{j=1}^{i} prob_j \tag{35}$$

Afterwards, a random number ranging from 0 to 1 is generated, the selected station is obtained through

$$s_k = \arg\min_{r_i \in R_k} |prob_i^{acc} - rand| \tag{36}$$

The selected station will be moved to the front of the route in R_k due to its relatively higher priority. The process repeated until certain amount of stations are selected and moved. Notice that here greedy search is not adopted, i.e., directly forcing the station with the highest priority to appear at front to keep the diversity of the solution because working time also makes up for the rebalancing cost, and is strictly related to the rebalancing demand of each station whereas prioritized stations are likely to hold more rebalancing demand compared to the other stations. Meanwhile, the roulette wheel is executed for only limited times, because only little fraction of the stations inside the service window will be visited given a rolling period. The procedure of the priority-based heuristic is given in Algorithm 3. Figure 5 gives an example of the priority-based heuristic. Station 1 is put in the front of the route due to its higher priority.

V. NUMERICAL STUDIES

We proceed our work to validate and test its performance on a real world instance by some numerical studies. All experiments are run on macOS Big Sur with 2.4GHz quad-core Intel Core i5 and 16 GB memory and all the algorithms are implemented using Python 3.7.

A. DATA DESCRIPTION

Nanjing launched the docked bike-sharing programs in January 2013. Supported by the government and as a non-profit project for citizens, Nanjing docked bike-sharing system has launched 1027 bike-sharing stations by the end of 2017. The BSS smart card dataset is provided by Nanjing Public Bicycle Company, involving BSS trips from 1^{st} Sep. 2017 to 30^{th} Sep, 2017. Each trip records user ID, bike ID, starting timestamps, starting longitude, starting latitude, ending timestamps, ending longitude and ending latitude. At data preprocessing stage, docked bike-sharing trips with the following properties have been removed:trip distance shorter than 100 m or longer than 5 km, as suggested by Shen *et al.* [50]; trip duration less than 30s or longer than 2h, as suggested by Pal and Zhang *et al.* [9]; trips without complete journey details.

Algorithm 3 Priority-Based Heuristic

Input: R: a sequential route encoding,
Output: R^{LS}: the route generated after heuristic search
decode the route into routes, each for one truck
$N \leftarrow$ the number of route decoded, i.e, the truck number
for $i \leftarrow 1$ to N **do**
  $rng_i \leftarrow$ the range of priority of route i by (33)
end for
$rt_1, rt_2 \leftarrow$ two routes with the smallest priority ranges
$rt_{tmp} \leftarrow rt_1 \cup rt_2$
$s_{h_1}, s_{h_2} \leftarrow$ two most prioritized stations from rt_{tmp}
$s_{l_1}, s_{l_2}, \leftarrow$ two least prioritized stations from r_{tmp}
$rt_1' \leftarrow rt_1 \setminus (s_{h_1} \cup s_{h_2} \cup s_{l_1} \cup s_{l_2}) \cup s_{h_1} \cup s_{l_1}$
$rt_2' \leftarrow rt_2 \setminus (s_{h_1} \cup s_{h_2} \cup s_{l_1} \cup s_{l_2}) \cup s_{h_2} \cup s_{l_2}$
$R' \leftarrow R \setminus (rt_1 \cup rt_2) \cup rt_1' \cup rt_2'$
if R' dominates R **then**
  $R^{LS} \leftarrow R'$
else
  $R^{LS} \leftarrow R$
end if
decode R^{LS} into routes
for $i \leftarrow 1$ to N **do**
  $s_k \leftarrow$ roulette wheel selected station based on priority for route i
  $rt_i \leftarrow$ move s_k to front of rt_i
end for
if R^{LS} dominates R **then**
  return R^{LS}
else
  return R
end if

In order to measure the effectiveness degree of bike-sharing stations, Station Turnover Rate (STS) is defined as follows:

$$STS_{return_i} = \frac{return_i}{C_i} \tag{37}$$

$$STS_{rent_i} = \frac{rent_i}{C_i} \tag{38}$$

STS_{return_i} is the ratio between the total number of returns $return_i$ at the end of the day at the station i and, the station capacity C_i. STS_{rent_i} can be calculated for the total number of rents ($rent_i$) at the end of the day at the station i. The higher the rate, the more effective the station is.

Results show that the average STS_{return_i} and STS_{rent_i} for the stations in urban area are 2.98 and 3.03 respectively,

TABLE 2. Information of the most prioritized stations in stage 1.

Station ID	number of bikes at 8:00	number of docks	predicted rent amount from 8:00-9:00	predicted return amount from 8:00-9:00	actual rent amount from 7:45-8:00	actual return amount from 7:45-8:00	variation rate (bike/min)	expected service time (min)	rebalancing demand	distance to depot (km)	station priority
15067	0	36	6	3	5	2	0.12	0	15	2.79	2.67
13094	1	35	4	6	5	1	0.12	8.33	14	1.98	2.62
11024	1	47	2	2	4	1	0.1	10	15	4.3	2.58
15031	7	27	21	14	11	3	0.33	21.21	19	2.11	2.53
15077	2	27	11	15	6	2	0.1	20	10	2.32	2.49
13097	4	26	11	14	8	2	0.18	22.22	12	1.86	2.44
13017	3	36	6	2	3	0	0.13	23.08	12	2.31	2.40
15075	7	33	18	15	8	1	0.26	26.92	16	0.99	2.36
15029	24	60	54	19	19	4	0.79	30.38	36	0.82	2.32
15056	2	45	3	2	3	1	0.07	28.57	12	3.42	2.27
11127	5	37	13	9	5	1	0.17	29.41	13	4.52	2.23
13025	21	58	58	25	17	6	0.64	32.81	29	3.79	2.19
15088	5	44	9	2	4	1	0.16	31.25	14	2.53	2.16
13095	1	22	8	9	2	1	0.03	33.33	6	3.2	2.12
11102	17	46	15	28	2	22	-0.77	37.66	-27	4.23	2.08
11126	11	29	17	6	8	2	0.29	37.93	13	2.59	2.04
11018	12	56	35	3	3	2	0.3	40	18	4.2	2.01
15001	23	52	34	4	9	0	0.55	41.82	21	3.04	1.97
11101	2	33	3	1	1	0	0.05	40	8	4.35	1.94
15003	20	58	48	29	12	3	0.46	43.48	20	2.11	1.90
11012	12	37	10	38	1	11	-0.57	43.86	-17	3.04	1.87
15007	10	58	29	19	7	3	0.22	45.45	15	0.51	1.84
13005	10	36	17	6	4	0	0.22	45.45	11	0.72	1.80
15048	25	35	59	57	8	15	-0.22	45.45	-11	2.3	1.77
13098	5	37	15	6	4	3	0.11	45.45	9	1.56	1.74
15023	6	41	7	9	5	1	0.12	50	10	0.86	1.71
15045	12	35	13	8	8	2	0.24	50	10	2.64	1.68
15062	3	52	4	2	1	0	0.05	60	11	1.68	1.65
11030	4	46	8	7	4	2	0.07	57.14	10	3.43	1.62
15012	17	60	35	14	5	3	0.24	60	10	1.41	1.59
15144	16	30	12	12	3	10	-0.23	60	-6	0.89	1.56
15009	12	55	9	14	5	1	0.09	60	5	0.51	1.54
13013	14	59	31	20	4	2	0.16	60	8	2.78	1.51
15143	10	35	11	20	5	1	0.06	60	1	0.62	1.48
15037	17	48	28	58	8	13	-0.42	60	-4	1.19	1.46
15047	10	41	17	11	4	2	0.12	60	6	2.22	1.43
15120	5	24	0	0	2	0	0.07	60	5	1.83	1.40
11021	12	34	30	18	8	5	0.2	60	7	3.99	1.38
11130	28	43	29	4	10	4	0.41	60	6	2.82	1.36
13009	11	51	13	15	1	0	0.02	60	1	1.2	1.33
15035	13	58	16	19	3	1	0.04	60	2	1.78	1.31
15100	12	39	17	13	2	0	0.1	60	2	1.88	1.28
15061	10	36	4	2	1	0	0.05	60	1	1.88	1.26
15068	28	59	26	11	8	2	0.33	60	4	2.59	1.24
13062	17	41	20	11	6	2	0.21	60	4	2.84	1.22
15038	12	34	14	9	3	1	0.11	60	2	2.38	1.20
11002	9	32	15	9	1	1	0.05	60	1	2.67	1.17
15042	11	47	26	44	8	2	0.05	60	2	2.88	1.15
15070	8	35	9	2	2	2	0.06	60	3	3.27	1.13
11011	7	32	16	38	8	2	0.02	60	1	3	1.11
15057	12	45	13	14	3	1	0.06	60	1	3.09	1.09
11104	14	28	10	14	9	2	0.2	60	4	4.67	1.07
11009	9	29	11	3	2	2	0.07	60	2	3.59	1.06
11028	4	31	15	64	7	6	-0.38	60	-2	4.12	1.04
11271	8	34	4	2	0	0	0.02	60	1	4.75	1.02
11035	10	45	3	1	1	1	0.02	60	1	5.04	1.00

which are higher than the stations outside the urban area (2.27 and 2.26 respectively). This is consistent with the finding of previous research [51], which found that the docked BSS in Nanjing has indicated significant imbalance of temporal and spatial demand of BSS trips in main urban area. Considerable efforts are needed to redistribute BSS to keep a high level of service quality. Therefore, in this research, 164 BSS stations within the 28 km^2 main urban area are selected. Figure 6 shows the BSS station distribution map of study area. Using the BSS smart card data and a short-term prediction model, the rental and return demand of shared bikes at the station level are predicted [52]. More details about the prediction model can be obtained in a study from [52].

B. EFFECTIVENESS OF PB-MOEA/D

In order to illustrate the effectiveness of the proposed Priority Based MOEA/D (PB-MOEA/D) algorithm for solving the multiobjective rebalancing problem, comparisons with Non-dominated sorting genetic algorithm II (NSGA-II) [53] and Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [45], two commonly used algorithms in solving multiobjective optimization problems [23] are conducted. First, the rebalancing demand is determined with the method proposed in this paper. The information of the stations in the first stage is shown in Table 2 sorted by priority. Note that according to the rolling horizon strategy, the demand and priority of the stations are dynamically altered and are

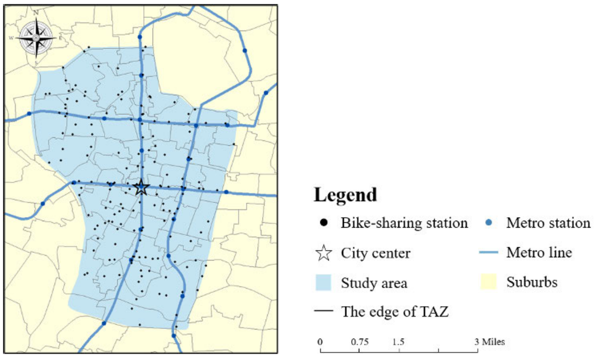


FIGURE 6. BSS station distribution map of study area.

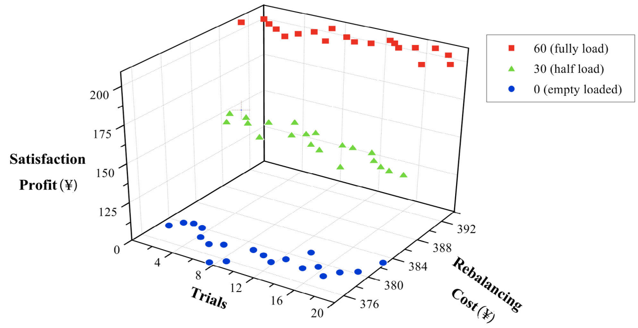


FIGURE 7. Satisfaction Profit and Cost with 3 different initial truckloads.

recalculated after every rolling period. Here we only list the information of the first stage.

The lengths of rebalancing horizon are uniformly set to 1 hour from 8:00 A.M. to 9:00 A.M. with 4 rolling stages of equal length (15 min each). The size of service window is set to be 1/2 the size of all stations with demand, i.e., the first 28 stations in Table 2 are selected to be optimized in the first stage. The distance of any two stations are assumed to be the Euclidean distance according to the coordinates of the stations. For the first stage, 3 trucks are used. After the first stage, due to the reduction of scale of the station in need of rebalancing, the truck with the least inventory quit service in the remaining stages and the other 2 trucks continue the rebalancing work until the end of the horizon. Three initial quantities of truck load when departing from the depot are compared, which includes 60 (fully loaded [54]), 30 (half loaded [26]) and 0 (empty loaded [55]). For each amount of truck load, 20 runs of the optimization using PB-MOEA/D are conducted, and the result is shown in Figure 7. Each point in the figure represents the highest satisfaction profit and lowest rebalancing cost of the certain trial. It can be observed that under our data, departing from the depot fully loaded could achieve the most user satisfaction profit, and since more users are satisfied, the rebalancing cost would rise from ¥378.17 (empty loaded) to ¥385.81 (half loaded) and then to ¥391.41 (fully loaded), but the growth is relatively slight compared to the satisfaction profit, which increases from ¥102.50 (empty loaded) to ¥154.37 (half loaded) and then to ¥206.77 (fully loaded). Therefore, we set the truck fully loaded when departing from the depot.

The parameter setting of the three algorithms and regarding rebalancing model are listed in Table 3 and the three algorithms were run under real world data described above.

For each stage, optimal solutions of routes are generated, and we choose the route with the best satisfaction profit to proceed to the next stage. The reason for this preference is that through extensive experiments, we notice that the increase of rebalancing cost is relatively trivial compared to its corresponding notable raise in satisfaction profit. Figure 8 shows the nondominated solutions with the lowest cost and highest satisfaction profit for four stages, respectively. The p-value of

TABLE 3. Experimental parameter settings.

Parameter	Value
Population Size	100 [56]
Termination Generation	250 [45]
Crossover Operator	Edge recombination [57]
Mutation Operator	Inversion mutation [58]
Crossover Rate	0.5 [59]
Mutation Rate	Reciprocal of the number of stations [59]
Number of weight vector neighbors	25 (MOEA/D, PB-MOEA/D) [56]
Neighbor mating probability	0.7 (MOEA/D, PB-MOEA/D) [59]
Cost of single run (ω_r)	¥40 / truck [60]
Loading and unloading cost (ω_{st})	¥0.42 / min [61]
Traveling cost for one car (ω_d)	¥0.67 / min [62]
Velocity of truck (v)	420 m / min [63]
Rolling period length (ζ)	15 min [64]
Maximum travel distance of a truck ($DMAX$)	35 km [65]

t-test [66] on the difference between the values on satisfaction profit generated by PB-MOEA/D and NSGA-II is 7.93E-07, and by PB-MOEA/D and MOEA/D is 1.09E-11, which are all less than 0.01, showing the significant differences between the outputs of the algorithms.

It can be observed that a small increase of 7.02% (¥29.2) on rebalancing cost can result in drastic growth in satisfaction profit of 1233.7% (¥288.7). However, this preference still depends on the operators of the rebalancing. If the BSS is operated by the government, the societal benefit measure, such as user satisfaction, should be prioritized in the objective function. However, if the BSS is operated by a private operator, the rebalancing cost would be the key concern [67].

Each algorithm was run 20 times, and for each algorithm, nondominated solutions for each run are collected into a solution set. For PB-MOEA/D, we set the local

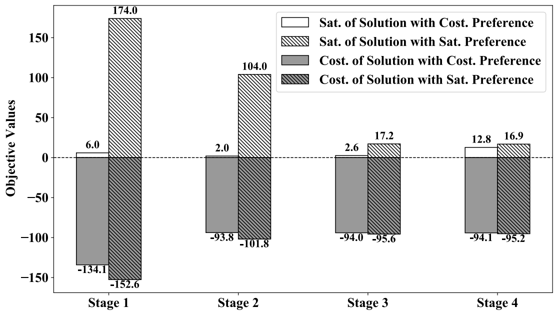


FIGURE 8. Nondominated Solutions of 4 rolling stages in different preferences.

TABLE 4. The best solutions & computational time of the three algorithms.

	NSGA-II		MOEA/D		PB-MOEA/D	
	Sat.	Cost.	Sat.	Cost.	Sat.	Cost.
Stage 1	147.0	126.4	162.1	143.5	174.0	134.1
Stage 2	37.1	87.4	103.8	94.5	104.0	93.8
Stage 3	12.5	88.5	17.2	94.1	17.2	94.0
Stage 4	12.7	87.0	16.9	94.4	16.9	94.1
Average CPU Time (sec.)	300.5		293.8		305.2	

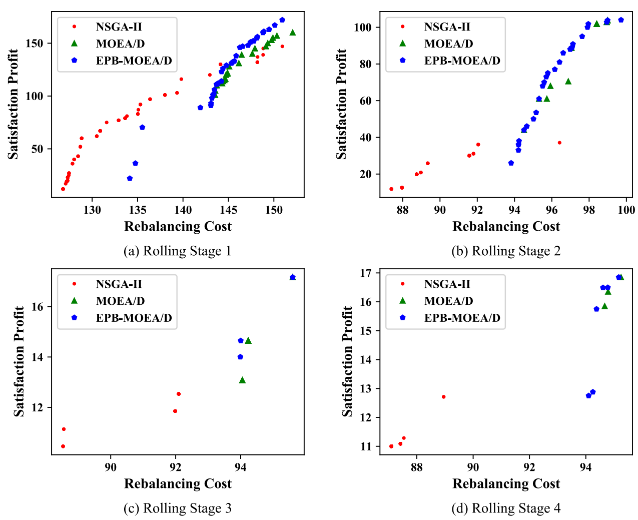


FIGURE 9. Nondominated Solutions of 4 rolling stages under three algorithms.

search limit as 10 times to limit the computational cost [56]. After 20 runs, we calculate the nondominated solution in the solution set of the algorithm to get the final nondominated solutions. We show the solutions with the highest satisfaction profit (*Sat.*) and lowest rebalancing cost (*Cost.*) respectively in Table 4 and the nondominated solutions of three algorithms are given in Figure 9. The results in Table 4 show that our proposed method outweighs NSGA-II and MOEA/D especially in achieving higher satisfaction profit. It is noticeable that in the first period, there are many stations waiting to be rebalanced, and the proposed algorithm shows better performance by a large margin. With the stage moving forward, the number of stations with rebalancing request decreases,

TABLE 5. Comparisons of indicators on satisfaction & truck travel distance.

	No Rebalancing	NSGA-II	MOEA/D	PB-MOEA/D
Actual Demand	574	574	574	574
Served Users	470	490	511	528
Unmet Demand	104	84	63	46
Served Demand Increase Rate	-	4.26%	8.72%	12.34%
Unmet Demand Reduction rate	-	19.23%	39.42%	55.77%
Distance traveled by trucks	-	27.88 km	26.02 km	25.52 km

TABLE 6. Set Coverage of the other two algorithms versus PB-MOEA/D.

	PB-MOEA/D vs MOEA/D	MOEA/D vs PB-MOEAD	PB-MOEA/D vs NSGA-II	NSGA-II vs PB-MOEA/D
Stage 1	0.79	0	0.49	0.43
Stage 2	0.29	0	0.25	0
Stage 3	0.33	0	0	0
Stage 4	0.5	0	0	0

and the corresponding results given by all algorithms become relatively closer. In terms of the average computational time, due to the time consumption of the priority-based local search, PB-MOEA/D consumes slightly more time for iteration of 250 generation compared to NSGA-II (4.7 seconds) and MOEA/D without local search (11.2 seconds), which is acceptable for achieving a better result. The amount of served users and unmet demand are also calculated for comparison, as is shown in Table 5. It can be observed that the system could serve more users after rebalancing with any of the three algorithms proposed, and using PB-MOEA/D could achieve the most served users and least unmet demand. Note that the served users increase rate and unmet demand reduction rate are all based on the value of the system with no rebalancing conducted. Moreover, using PB-MOEA/D could decrease the truck travelling distance comparing to MOEA/D and NSGA-II.

To evaluate the quality of Pareto solution set obtained from each algorithm, we calculate the set coverage [49] of nondominated solution set given by one algorithm over the other. Specifically, let R_1 and R_2 be two approximations to the Pareto front of a MOP; $C(R_1, R_2)$ is defined as the percentage of the solutions in R_2 that are dominated by at least one solution in R_1:

$$C(R_1, R_2) = \frac{|\{p \in R_2 \mid \exists q \in R_1 : q \ dominates \ p\}|}{|R_2|} \tag{39}$$

It is evident from Table 6 that in four stages, no solution obtained in MOEA/D without priority heuristics dominates any solutions in PB-MOEA/D. When it comes to NSGA-II, the superiority tends to be weaker, but PB-MOEA/D still performs better than NSGA-II. This is consistent with the result shown in Figure 9, that the nondominated solutions found by NSGA-II concentrate in the area of lower satisfaction. When focusing on satisfaction profit below 80 in stage 1 and 40 in stage 2, we could indeed observe some solutions discovered by NSGA-II dominate PB-MOEA/D. Overall, our

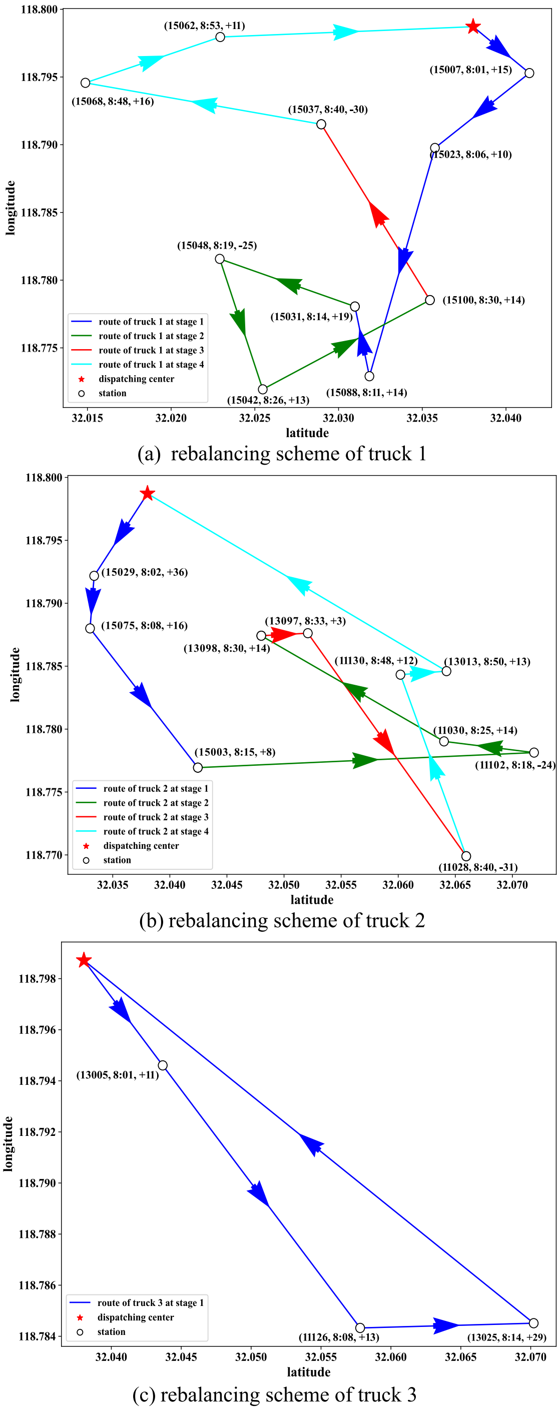


FIGURE 10. Rebalancing schemes of 3 trucks.

experiment shows that PB-MOEA/D outperforms the other two algorithms in finding solutions with better quality.

Figure 10 shows the rebalancing schemes of the 3 trucks including the routes and rebalancing amount at each station. Different colors of routes indicate different rolling stages, and the time and rebalancing amount at each station are marked in the figure as (*station ID*, *time*, *amount*).

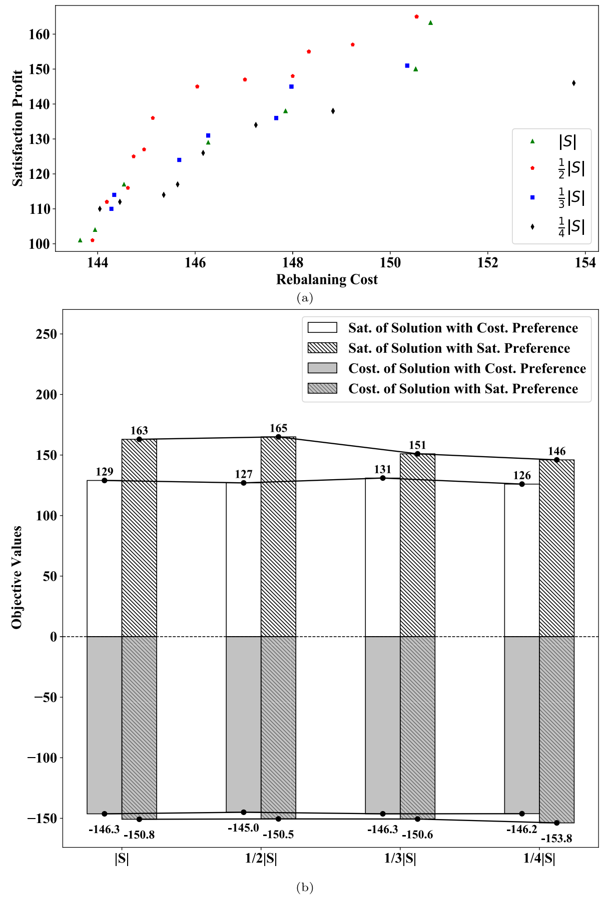
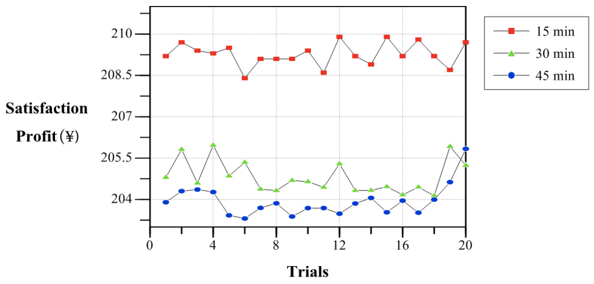


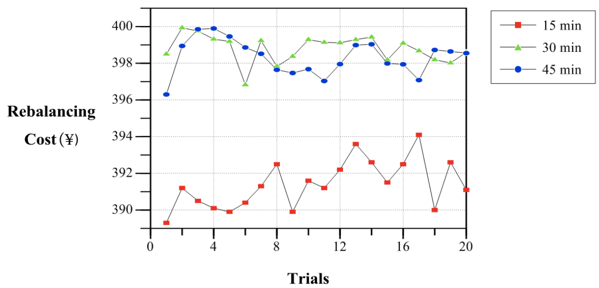
FIGURE 11. Optimization results under different sizes of service window.

C. EFFECTIVENESS OF ROLLING STAGE WINDOW SIZE

To decompose the problem and reduce the scale of the search space, certain prioritized stations are selected into the service window of each rolling period. If the window size is relatively large, the search space grows, and the solution diversity will correspondingly increase while the solution quality gets decreased. However, windows sizes being too small could prevent the discovery of some solutions due to limited search space. Therefore, experiments of four scales of window size, namely $|S|$, $\frac{1}{2}|S|$, $\frac{1}{3}|S|$ and $\frac{1}{4}|S|$ are conducted for comparison, where $|S|$ represents the number of stations. The result of the optimal solutions using PB-MOEA/D under different window size for stage 1 are illustrated in Figure 11(a). Figure 11(b) gives the solution with the highest satisfaction profit and lowest rebalancing cost and compared in different window scale respectively. Figure 11 shows that the window size of $\frac{1}{2}|S|$ leads to the best result compared to other three scales. It is worth mentioning that for real-world application and based on the analysis of coherent reversing relationship between the two objectives above, we set a threshold of the satisfaction of solution to be 120, i.e., only solutions with satisfaction profit over 120 are compared.



(a) satisfaction profit under different rolling period lengths



(b) rebalancing cost under different rolling period lengths

FIGURE 12. Optimization results under different sizes of service window.

D. EFFECTIVENESS OF THE ROLLING PERIOD LENGTH

Apart from the rolling window size, we also investigate the influence of rolling period length which is closely related to the update frequency of the rebalancing system. Experiments of three length of the rolling period, namely 15min, 30min and 45min are conducted with 20 runs for comparison. The overall result of the optimal solutions using PB-MOEA/D under different rolling period length are demonstrated in Figure 12. Figure 12 shows that the length of 15 min leads to the best satisfaction profit and rebalancing cost compared to other three scales.

VI. CONCLUSION

This paper focuses on modeling a dynamic BSS rebalancing problem with maximum satisfaction profit and minimum rebalancing costs, attempting to provide optimal routes under real-world instances. An approach using both historical and predicted rental and return demand is proposed to dynamically determine the rebalancing demand, which could avoid a second time rebalancing for the same station. Refinement of satisfaction is made by differentiating the satisfaction of each bike scheduled to the station according to the specific time a bike is delivered or picked up. Moreover, the satisfaction profit is defined to assess the rebalancing service more scientifically, which takes both timeliness and the amount of satisfied demand into account. A novel priority-based heuristic was introduced in the MOEA/D algorithm to solve the problem, which intends to improve the solution through local search and further improve the performance of the algorithm. Numerical studies are conducted to validate the effectiveness of the proposed model and algorithm. The results show that the proposed PB-MOEA/D algorithm outperforms the MOEA/D without proposed heuristic search, as well as NSGA-II. The result shows PB-MOEA/D outweighs NSGA-II and MOEA/D in every stage. The results also demonstrate that slightly expanding the rebalancing cost can bring significant growth in satisfaction profit. The experimental results also demonstrates that the best optimization performance can be achieved with a service window of half the size of the station amount and the rolling length of 15 minutes.

Future studies could be carried out around the extension of our current model. First, the model works for the dynamic rebalancing of docked BSSs, and it cannot be directly applied to dockless BSSs. However, it can be extended for dynamically rebalancing the dockless BSSs if more conditions are considered: 1) unlike the docked BSSs, there exist no fixed stations in dockless BSSs. Therefore, the virtual stations of dockless BSSs should be generated in the first place; 2) scattered dockless shared bikes should be collected to virtual stations, which takes more time and costs than rebalancing the docked BSSs; 3) lacking the protection and maintenance like docked BSSs, dockless shared bikes, including their necessary parts such as the original QR codes, seats, pedals, are deliberately damaged [68]. The collection of malfunctioning shared bikes should be considered when rebalancing the dockless BSSs. The number of trucks, as well as the speed of trucks influence the rebalancing cost to a large extent, thus considering the factors regarding the rebalancing vehicles in the model is also worth researching. Besides, the coefficient for bike inventory variation rate directly affects the demand evaluation, and the maximum satisfaction profit and is a vital factor in weighing the rebalancing cost and satisfaction. The analysis of these importance values should be considered in further research.

REFERENCES

[1] J. Dill, "Bicycling for transportation and health: The role of infrastructure," *J. Public Health Policy*, vol. 30, no. S1, pp. S95–S110, Jan. 2009.

[2] M. Dell'Amico, M. Iori, S. Novellani, and A. Subramanian, "The bike sharing rebalancing problem with stochastic demands," *Transp. Res. B, Methodol.*, vol. 118, pp. 362–380, Dec. 2018.

[3] Y. Liu, W. Szeto, and S. C. Ho, "A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 208–242, Jul. 2018.

[4] A. Pal, Y. Zhang, and C. Kwon, *Analyzing Mobility Patterns and Imbalance of Free Floating Bike Sharing Systems*. Tampa, FL, USA: Univ. South Florida, 2017.

[5] F. Soriguera and E. Jiménez-Meroño, "A continuous approximation model for the optimal design of public bike-sharing systems," *Sustain. Cities Soc.*, vol. 52, Jan. 2020, Art. no. 101826.

[6] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 1393–1400, Jul. 2019.

[7] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing users for balancing bike sharing systems," in *Proc. AAAI*, 2015, pp. 723–729.

[8] Y. Lu, U. Benlic, and Q. Wu, "An effective memetic algorithm for the generalized bike-sharing rebalancing problem," *Eng. Appl. Artif. Intell.*, vol. 95, Oct. 2020, Art. no. 103890.

[9] A. Pal and Y. Zhang, "Free-floating bike sharing: Solving real-life large-scale static rebalancing problems," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 92–116, Jul. 2017.

[10] I. A. Forma, T. Raviv, and M. Tzur, "A 3–step math heuristic for the static repositioning problem in bike–sharing systems," *Transp. Res. B, Methodol.*, vol. 71, pp. 230–247, Jan. 2015.

[11] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld, "Dynamic lookahead policies for stochastic–dynamic inventory routing in bike sharing systems," *Comput. Oper. Res.*, vol. 106, pp. 260–279, Jun. 2019.

[12] L. Caggiani and M. Ottomanelli, "A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems," *Procedia Social Behav. Sci.*, vol. 87, pp. 203–210, Oct. 2013.

[13] L. Caggiani, R. Camporeale, M. Ottomanelli, and W. Y. Szeto, "A modeling framework for the dynamic management of free-floating bike-sharing systems," *Transp. Res. C, Emerg. Technol.*, vol. 87, pp. 159–182, Feb. 2018.

[14] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl, "Balancing bicycle sharing systems: An approach for the dynamic case," in *Evolutionary Computation in Combinatorial Optimisation* (Lecture Notes in Computer Science), vol. 8600, C. Blum and G. Ochoa, Eds. Berlin, Germany: Springer, 2014, pp. 73–84.

[15] E. O'Mahony and D. B. Shmoys, "Data analysis and optimization for (CITI) bike sharing," in *Proc. 29th AAAI Conf. Artif. Intell. AAAI*, 2015, pp. 687–694.

[16] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A bike-sharing optimization framework combining dynamic rebalancing and user incentives," *ACM Trans. Auto. Adapt. Syst.*, vol. 14, no. 3, pp. 1–30, Mar. 2020.

[17] B. Legros, "Dynamic repositioning strategy in a bike-sharing system; How to prioritize and how to rebalance a bike station," *Eur. J. Oper. Res.*, vol. 272, no. 2, pp. 740–753, Jan. 2019.

[18] E. Lee, B. Son, and Y. Han, "Optimal relocation strategy for public bike system with selective pick-up and delivery," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2674, no. 4, pp. 325–336, Apr. 2020.

[19] J. Zhu, W. Xu, H. Sun, and H. Zhu, "Research on optimization of shared bicycle real-time multimedia interactive dispatching system for intelligent transportation system," *Multimedia Tools Appl.*, pp. 1–17, Mar. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s11042-019-7415-9, doi: 10.1007/s11042-019-7415-9.

[20] M. Dell'Amico, M. Iori, S. Novellani, and T. Stützle, "A destroy and repair algorithm for the bike sharing rebalancing problem," *Comput. Oper. Res.*, vol. 71, pp. 149–162, Jul. 2016.

[21] C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a Dynamic Public Bike-Sharing System*, vol. 4. Montreal, QC, Canada: Cirrelt Montreal, 2012.

[22] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld, "Short-term Strategies for stochastic inventory routing in bike sharing systems," *Transp. Res. Procedia*, vol. 10, pp. 364–373, Jan. 2015.

[23] H. Jia, H. Miao, G. Tian, M. Zhou, Y. Feng, Z. Li, and J. Li, "Multiobjective bike repositioning in bike-sharing systems via a modified artificial bee colony algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 909–920, Apr. 2020.

[24] L. Caggiani and M. Ottomanelli, "A modular soft computing based method for vehicles repositioning in bike-sharing systems," *Procedia Social Behav. Sci.*, vol. 54, pp. 675–684, Oct. 2012.

[25] P.-S. You, "A two-phase heuristic approach to the bike repositioning problem," *Appl. Math. Model.*, vol. 73, pp. 651–667, Sep. 2019.

[26] R. Regue and W. Recker, "Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem," *Transp. Res. E, Logistics Transp. Rev.*, vol. 72, pp. 192–209, Dec. 2014.

[27] C. Shui and W. Szeto, "Dynamic green bike repositioning problem—A hybrid rolling horizon artificial bee colony algorithm approach," *Transp. Res. D, Transp. Environ.*, vol. 60, pp. 119–136, May 2018.

[28] L. Di Gaspero, A. Rendl, and T. Urli, "Constraint-based approaches for balancing bike sharing systems," in *Principles and Practice of Constraint Programming*, vol. 8124, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and C. Schulte, Eds. Berlin, Germany: Springer, 2013, pp. 758–773.

[29] A. A. Kadri, K. Labadi, and I. Kacem, "An integrated Petri net and GA–based approach for performance optimisation of bicycle sharing systems," *Eur. J. Ind. Eng.*, vol. 9, no. 5, p. 638, 2015.

[30] M. Rainer-Harbach, P. Papazek, B. Hu, and G. R. Raidl, "Balancing bicycle sharing systems: A variable neighborhood search approach," in *Proc. Evolutionary Computation in Combinatorial Optimization*, vol. 7832, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. P. Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. Middendorf, and C. Blum, Eds. Berlin, Germany: Springer, 2013, pp. 121–132.

[31] A. A. Kadri, I. Kacem, and K. Labadi, "A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems," *Comput. Ind. Eng.*, vol. 95, pp. 41–52, May 2016.

[32] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *IIE Trans.*, vol. 45, no. 10, pp. 1077–1093, Oct. 2013.

[33] D. Zhang, C. Yu, J. Desai, H. Lau, and S. Srivathsan, "A time-space network flow approach to dynamic repositioning in bicycle sharing systems," *Transp. Res. B, Methodol.*, vol. 103, pp. 188–207, Sep. 2017.

[34] W. Szeto, Y. Liu, and S. C. Ho, "Chemical reaction optimization for solving a static bike repositioning problem," *Transp. Res. D, Transp. Environ.*, vol. 47, pp. 104–135, Aug. 2016.

[35] M. Dell'Amico, E. Hadjicostantinou, M. Iori, and S. Novellani, "The bike sharing rebalancing problem: Mathematical formulations and benchmark instances," *Omega*, vol. 45, pp. 7–19, Jun. 2014.

[36] L. Di Gaspero, A. Rendl, and T. Urli, "A hybrid ACO+CP for balancing bicycle sharing systems," in *Hybrid Metaheuristics*, vol. 7919, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. J. Blesa, C. Blum, P. Festa, A. Roli, and M. Sampels, Eds. Berlin, Germany: Springer, 2013, pp. 198–212.

[37] K. Mellou and P. Jaillet, "Dynamic resource redistribution and demand estimation: An application to bike sharing systems," *SSRN Electron. J.*, pp. 1–58, Jan. 2019.

[38] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld, "Inventory routing for bike sharing systems," *Transp. Res. Procedia*, vol. 19, pp. 316–327, Jan. 2016.

[39] O. E. Daniel, "Smarter tools for (CITI) bike sharing," Ph.D. dissertation, Dept. Comput. Sci., Cornell Univ., New York, NY, USA, 2015.

[40] R. Salih-Elamin and H. Al-Deek, "Utilizing node importance to maximize locations coverage in a bike share system network," *Adv. Transp. Stud.*, vol. 51, pp. 121–140, Jul. 2020.

[41] W. Huang, B. Shuai, Y. Sun, Y. Wang, and E. Antwi, "Using entropy-TOPSIS method to evaluate urban rail transit system operation performance: The China case," *Transp. Res. A, Policy Pract.*, vol. 111, pp. 292–303, May 2018.

[42] Z. Ming, "Optimization of bike sharing scheduling problem based on data driven methodology," M.S. thesis, School Maritime Econ. Manage., Dalian Univ. Technol., Dalian, China, 2019.

[43] C.-L. Huang, S.-Y. Huang, W.-C. Yeh, and J. Wang, "Fuzzy system and time window applied to traffic service network problems under a multi-demand random network," *Electronics*, vol. 8, no. 5, p. 539, May 2019.

[44] Y. Jia, C. Wang, and L. Wang, "A rolling horizon procedure for dynamic pickup and delivery problem with time windows," in *Proc. IEEE Int. Conf. Autom. Logistics*, Aug. 2009, pp. 2087–2091.

[45] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[46] M.-G. Gong, L.-C. Jiao, D.-D. Yang, and W.-P. Ma, "Research on evolutionary multi-objective optimization algorithms," *J. Softw.*, vol. 20, no. 2, pp. 271–289, Apr. 2009.

[47] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[48] U. Baumgartner, C. Magele, and W. Renhart, "Pareto optimality and particle swarm optimization," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1172–1175, Mar. 2004.

[49] A. Konstantinidis, S. Pericleous, and C. Charalambous, "MOEA/D for a Tri–objective vehicle routing problem," in *Artificial Intelligence Applications and Innovations*, vol. 412, H. Papadopoulos, A. S. Andreou, L. Iliadis, and I. Maglogiannis, Eds. Berlin, Germany: Springer, 2013, pp. 131–140.

[50] Y. Shen, X. Zhang, and J. Zhao, "Understanding the usage of dockless bike sharing in singapore," *Int. J. Sustain. Transp.*, vol. 12, no. 9, pp. 686–700, Oct. 2018.

[51] M. Hua, X. Chen, S. Zheng, L. Cheng, and J. Chen, "Estimating the parking demand of free-floating bike sharing: A journey-data-based study of Nanjing, China," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118764.

[52] Y. Jin, X. Ma, Y. Yuan, and M. He, "Short-term prediction of bike-sharing demand using multi-source data: An attention-based graph convolution LSTM approach," in *Proc. 100th Transp. Res. Board Annu. Meeting*, Washington, DC, USA, to be published.

[53] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[54] C. Kloimüllner and G. R. Raidl, "Full-load route planning for balancing bike sharing systems by logic-based benders decomposition," *Networks*, vol. 69, no. 3, pp. 270–289, May 2017.

[55] Q. Tang, Z. Fu, and M. Qiu, "A bilevel programming model and algorithm for the static bike repositioning problem," *J. Adv. Transp.*, vol. 2019, pp. 1–19, Jun. 2019.

[56] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA–II on the multi-objective travelling salesman problem," in *Multi-Objective Memetic Algorithms*, vol. 171, C.-K. Goh, Y.-S. Ong, and K. C. Tan, Eds. Berlin, Germany: Springer, 2009, pp. 309–324.

[57] L. D. Whitley, T. Starkweather, and D. Fuquay, "Scheduling problems and traveling salesmen: The genetic edge recombination operator," in *Proc. ICGA*, 1989, pp. 133–140.

[58] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybern. Syst.*, vol. 24, no. 1, pp. 27–36, Jan. 1993.

[59] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[60] X. Cao, "Research on operating characteristics and dynamic scheduling optimization of public bicycle based on ic card data," M.S. thesis, School Transp., Southeast Univ., Nanjing, China, 2016.

[61] H. Jin, "Research on scheduling of shared bicycles based on mobike travel data," M.S. thesis, School Econ. Manage., Beijing Jiaotong Univ., Beijing, China, 2019.

[62] T. Ma, "Study on optimization of shared bicycle connection with urban rail stations," M.S. thesis, School Civil Transp. Eng., Beijing Univ. Civil Eng. Archit., Beijing, China, 2018.

[63] X. Zhang, "Research on scheduling problem of transport vehicles for shared bicycle," M.S. thesis, Transp. College, Jilin Univ., Changchun, China, 2019.

[64] P. Feng, "Research on scheduling management optimization method of public bicycle system," Ph.D. dissertation, School Transp., Southeast Univ., Nanjing, China, 2018.

[65] J. Kong, "Research on site demand prediction and repositioning optimization of free-floating bike sharing," M.S. thesis, School Automobile, Chang'an Univ., Xi'an, China, 2018.

[66] S. Datner, T. Raviv, M. Tzur, and D. Chemla, "Setting inventory levels in a bike sharing network," *Transp. Sci.*, vol. 53, no. 1, pp. 62–76, 2017.

[67] S. C. Ho and W. Szeto, "Solving a static repositioning problem in bike-sharing systems using iterated tabu search," *Transp. Res. E, Logistics Transp. Rev.*, vol. 69, pp. 180–198, Sep. 2014.

[68] X. Ma, Y. Yuan, N. Oort, Y. Ji, and S. Hoogendoorn, "Understanding the difference in travel patterns between docked and dockless bike-sharing systems: A case study in Nanjing, China," in *Proc. 98th Transp. Res. Board Annu. Meeting*, Washington, DC, USA, 2019, pp. 13–17.

RUNQIU HU received the bachelor's degree from the School of Computer, Nanjing University of Posts and Telecommunications, in 2018. He is currently pursuing the master's degree with the School of Computer Science and Engineering, Southeast University. His research interests include knowledge representation and reasoning, bike-sharing system optimization, and multiobjective optimization.

ZHIZHENG ZHANG received the Ph.D. degree from Southeast University, in 2006. He is currently an Associate Professor and a M.S. Supervisor with the School of Computer Science and Engineering, Southeast University. His research interests include preference logic, inductive logic programming and reasoning under uncertain, and so on.

XINWEI MA received the M.S. degree in transportation planning and management from the Hebei University of Technology, in 2013, and the Ph.D. degree in transportation engineering from the School of Transportation, Southeast University, in 2020. His research interests include public transport shared mobility, transport data analytics, and intelligent transport systems.

YUCHUAN JIN received the bachelor's degree in traffic engineering from Southeast University, in 2019. He is currently pursuing the master's degree in transport science with the Royal Institute of Technology (KTH). His research interests include transportation, big data analysis and modeling, machine learning, data mining, and intelligent transportation systems.