

Research Article

The Biobjective Bike-Sharing Rebalancing Problem with Balance Intervals: A Multistart Multiobjective Particle Swarm Optimization Algorithm

Yongji Jia , Yuanyuan Xu, Dong Yang, and Jia Li

Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China

Correspondence should be addressed to Yongji Jia; yjjia@dhu.edu.cn

Received 14 June 2020; Revised 2 September 2020; Accepted 7 September 2020; Published 17 September 2020

Academic Editor: Kailong Liu

Copyright © 2020 Yongji Jia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The bike-sharing system (BSS), as a sustainable way to deal with the “last mile” problem of mass transit systems, is increasingly popular in recent years. Despite its success, the BSS tends to suffer from the mismatch of bike supply and user demand. BSS operators have to transfer bikes from surplus stations to deficit stations to redistribute them among stations by means of trucks. In this paper, we deal with the bike-sharing rebalancing problem with balance intervals (BRP-BIs), which is a variant of the static bike-sharing rebalancing problem. In this problem, the equilibrium of station is characterized by a balance interval instead of a balance point in the literature. We formulate the BRP-BI as a biobjective mixed-integer programming model with the aim of determining both the minimum cost route for a single capacitated vehicle and the maximum average rebalance utility, an index for the balanced degree of station. Then, a multistart multiobjective particle swarm optimization (MS-MOPSO) algorithm is proposed to solve the model such that the Pareto optimal solutions can be derived. The proposed algorithm is extended with crossover operator and variable neighbourhood search to enhance its exploratory capability. Compared with Hybrid NSGA-II and MOPSO, the computational experimental results demonstrate that our MS-MOPSO can obtain Pareto optimal solutions with higher quality.

1. Introduction

Nowadays, with the increasing concerns regarding global warming, countries around the world have begun to save energy and reduce carbon emissions. In order to achieve the low-carbon and sustainable transport, bike-sharing systems (BSSs) are increasingly popular, as they can reduce travel costs, road congestion, and greenhouse gas emissions [1]. The first BSS was introduced in Amsterdam in 1965, and then its popularity increased as a new short-distance transport mode in the last two decades across the globe [2, 3].

Despite its success, the BSS tends to suffer from the mismatch of bike supply and user demand. This happens when a user arrives at a station that has no bike available or that has no parking slot available. As can be seen from Figure 1, there are evident morning and evening peaks in the bike-sharing usage on workdays. During the morning rush

hour, users pick up bikes in residential areas, make a one-way point to point transportation, and return them in working areas. This morning peak leads to excessive bikes in working areas as well as insufficient bikes in residential areas, depicted in Figure 2, and vice versa during the evening rush hour. This mismatch results in poor service level and low customer satisfaction, for the BSS has little chance to complete self-balance in a short time.

To improve the service level, it is necessary for the BSS operator to realize the rebalance between bike supply and user demand by transporting bikes from surplus stations to deficient stations with the help of trucks, referred to as the bike-sharing rebalancing problem (BRP) [4]. BRPs can be divided into static problems and dynamic ones according to whether user activity is negligible or not.

In this paper, we focus on the static bike-sharing rebalancing problem with balance intervals (BRP-BIs), where the equilibrium of station is characterized by balance

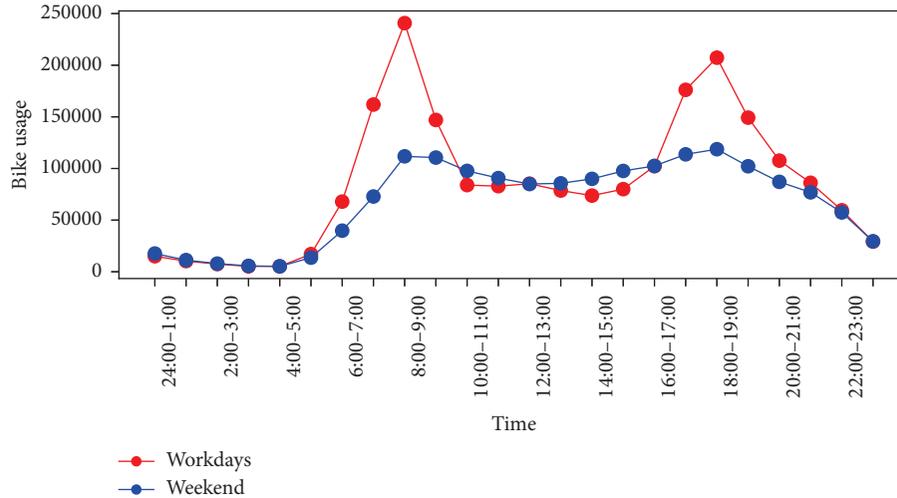


FIGURE 1: Bike-sharing usage on workdays and weekends.

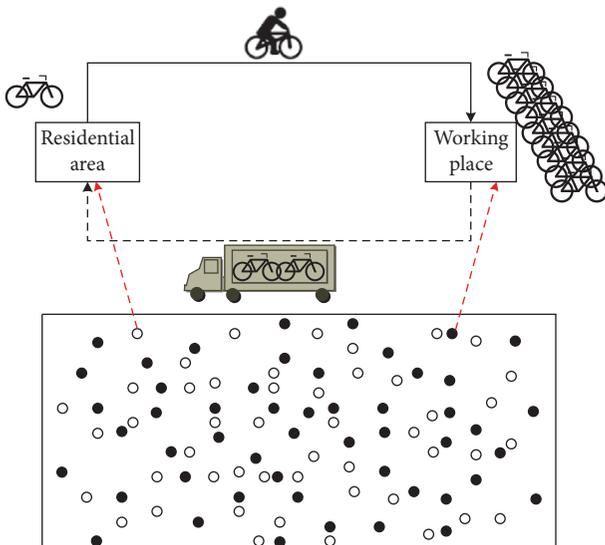


FIGURE 2: The mismatch phenomenon of a BSS during the morning peak. Black circles represent stations with excessive bikes, while white circles represent stations with insufficient bikes.

interval instead of balance point in the literatures. And then, we formulate this problem as a biobjective mixed-integer programming model with objectives of minimizing the total travel distance as well as maximizing the average rebalance utility, an index for the balanced degree at each station. The definition and description of balance interval and rebalance utility are described in Section 3.

BRP-BI, as a variant of one-commodity pickup-and-delivery traveling salesman problem (1-PDTSP) [5], is a complex problem that includes decisions regarding the route that the vehicle should follow and the number of bikes that should be loaded or unloaded at each station. It is more complicated than traveling salesman problem, a well-known NP-hard problem, since the numbers of bikes loaded or unloaded are decision variables. Besides, BRP-BI is based on the balance interval, and it has two conflicting objectives. It

is therefore more complex and more difficult to be solved exactly. As the problem size increases, the calculation time will increase exponentially. Thus, the real-life BRP-BI instances cannot be solved exactly within short computation time.

To handle this biobjective NP-hard problem in a runtime acceptable to BSS operators, we develop the multistart multiobjective particle swarm optimization (MS-MOPSO) algorithm. MOPSO has been found to be successful to solve a wide variety of multiobjective optimization problems, but until now, it has not been extended to deal with biobjective BRP-BI. MS-MOPSO is thus proposed to obtain the Pareto optimal solutions for this biobjective problem. To avoid falling into the local optimum and to enhance the exploratory capability, crossover operator and variable neighborhood search are incorporated to enable MS-MOPSO to achieve balanced exploration and exploitation capabilities.

The main contributions of this paper lie on the following:

- (i) To introduce a new and practical concept of balance interval, which is different from balance point in the literatures, and then, BRP-BI is presented
- (ii) To construct a biobjective mixed-integer programming model to formulate this problem
- (iii) To develop a MS-MOPSO enhanced by the crossover operator and variable neighborhood search to deal with this biobjective model

The remainder of this paper is organized as follows: Section 2 summarizes the literature review on the related problem. Section 3 formally proposes the concept of balance interval and rebalance utility and then the biobjective mixed-integer programming model for BRP-BI. Section 4 develops the MS-MOPSO to obtain the nondominated optimal solutions for BRP-BI. Section 5 contains the computational experiments and sensitivity analysis. Finally, Section 6 draws some conclusions about this work.

2. Literature Review

During the past decade, BRPs have attracted the considerable attention in the literatures. Generally speaking, they can be roughly classified into two categories: static and dynamic, depending on whether the user activity of a BSS is negligible or not. If the user activity is low and has no effect on the rebalancing operations (such as during the midnight), it is static BRP (SBRP). Otherwise, it is dynamic BRP (DBRP) [6]. In the following, we will discuss the literatures related to the SBRP and DBRP, respectively.

2.1. Static Bike-Sharing Rebalancing Problem. Many approaches to solve the SBRP revolve around optimization models that maximize profit or minimize travel costs. Dell'Amico et al. [7] provide four mixed-integer programming models with the objective of minimizing total costs, where a fleet of capacitated vehicles is employed to redistribute the bikes. The same objective is presented by Zhang et al. [8], who propose a hybrid discrete PSO algorithm to solve it. Erdoğan et al. [5] and Cruz et al. [9] solve the same model with a single vehicle, where multiple visits are allowed at each station. Duan et al. [10] focus on a vehicle routing problem in the rebalancing operations with the objective of minimizing the total driving distance, and a greedy algorithm is proposed. Shi et al. [11] propose an improved PSO algorithm to minimize the total costs including the fixed costs and the variable transport costs. In addition, some researchers have taken time into consideration in the SBRP models. Casazza et al. [12] incorporate time into the constraints such that each route does not exceed time limitation. In the study of Bulhões et al. [3], service time is limited and the objective is to minimize the travel time.

All the problems above need to fully meet the rebalance demands, but other researchers are trying to relax these constraints. Papazek et al. [13] set the primary goal to minimize the absolute deviation between target and final fill levels for all stations. Gaspero et al. [14] solve the problem with the aim of minimizing the weighted sum of the total travel time and the total absolute deviation from the target number of bikes, while Raviv and Kolka [15] take it as a penalty cost. Szeto et al. [16] and Espegren et al. [17] also consider the same factors, but the objective is to minimize the weighted sum of unmet customer demand and operational time. Meanwhile, Ho and Szeto [18] define the objective as the sum of the penalty cost incurred at each station and the weighted total travel time. Raviv et al. [19] define the goal of the problem as minimizing the total cost of the system consisting of the weighted sum of the penalties incurred at all stations and the total operating costs. Ho and Szeto [20] present an iterated tabu search heuristic process to solve the SBRP with the goal of minimizing the total penalties incurred at all stations. Li et al. [21] consider solving a multitype bike rebalancing problem, which is formulated as a mixed-integer programming problem, aiming at minimizing the total costs consisting of the travel cost, the penalties due to unmet demands, and the penalties associated with the substitution and occupancy strategies. Broken

bikes are considered by Wang and Szeto [22] with the objective of minimizing the total CO₂ emissions of all vehicles. Usama et al. [23] also consider replacing faulty bikes in the system with the following objectives: user dissatisfaction to unmet demand, vehicle routing costs including handling, loading, unloading, and traveling cost, and the waiting time of repositioning vehicles.

2.2. Dynamic Bike-Sharing Rebalancing Problem. Due to the participations of users, there is a great deal of uncertainties in the BSS during the daytime and the SBRP models have failed to meet the actual dynamic rebalance demands. Therefore, some scholars have turned their attention to the DBRP under uncertainty circumstances.

Some researchers deal with uncertainty through predictions. Alvarez-Valdes et al. [2] estimate the unsatisfied demand and solve the problem with the objective of minimizing the global expected dissatisfaction for each possible number of bicycles at the beginning of the period. And then they use these estimates to guide redistribution algorithms. Zhang et al. [24] propose a dynamic bicycle repositioning methodology that considers inventory-level forecasting, user arrivals forecasting, bicycle repositioning, and vehicle routing in a unified manner. A multicommodity time-space network flow model is presented with the objective of minimizing the total vehicle travel costs plus the expected user dissatisfaction in the system. Caggiani et al. [25] propose a new comprehensive methodology that starts from the prediction of the number and position of bikes and ends with a relocation decision support system, aiming at achieving a high degree of user satisfaction and keeping the vehicle repositioning costs as low as possible.

In addition, other researchers embody uncertainty through stochastic demand based on scenarios. In the study of Yan et al. [26], the objective is to maximize the expected profit of the system, that is, the expected total bike rental income on the demand arcs minus the expected total bike holding cost on the holding arcs over all scenarios. Dell'Amico et al. [27] develop the stochastic programming model with the objective of minimizing the travel cost and the penalty costs for unfulfilled demands. Maggioni et al. [28] propose the two-stage and multistage stochastic optimization models to determine the optimal number of bikes to assign to each station at the beginning of the service. The objective is to minimize the sum of the procurement cost for the assigned bikes, the expected stock-out cost for shortage, the expected time-waste cost for overflow, and the expected transshipment cost for repositioning bikes at the end of the service. Yuan et al. [29] present a mixed-integer programming model with the objective of minimizing the daily costs including the capital cost and the expected operating cost.

Furthermore, there are still some studies dealing directly with dynamic factors. Legros [30] focuses on dynamic repositioning strategy in a BSS with the objective of minimizing the rate of arrival of unsatisfied users and solves it by a Markov decision process approach. You [31] develops a constrained mathematical model to deal with a multivehicle BRP, aiming to develop dynamic decisions to minimize the

sum of the transportation costs and unmet costs under service-level constraints over a planning horizon. Cao et al. [32] propose a station location problem under a zoning scheme in consideration of multiperiod demand, and the objective of this problem is to minimize the fixed construction costs and variable operational costs. Warrington and Ruchti [33] present a novel nested-flow formulation of the dynamic problem to minimize a generalized cost function, which includes penalty costs for unserved customer demands, costs for the movements of vehicles, and costs of loading and unloading. Regue and Recker [34] propose a framework to solve the DBRP based on four models: a demand forecasting model, a station inventory model, a redistribution needs model, and a vehicle-routing model. Their approach is proactive instead of reactive, as bike repositioning occurs before inefficiencies are observed.

2.3. Summary. Most studies directly give out the unbalanced demands in their BRP models without involving the concept of equilibrium of station. According to the equilibrium of station, model objective, and solution approach, Table 1 summarizes the literature considering the equilibrium of station.

In terms of equilibrium of station, the majority of these studies regard it as a balance point, but only two studies [5, 35] begin to pay attention to the concept of interval. Obviously, the balance point is a special case of the balance interval. The advantage of the balance interval is that it can decrease the number of rebalancing operations, thereby reducing the complexity of the BRP model. In the study of Erdoğan et al. [5], demand interval is proposed to refer to an inventory of bicycles lying between given lower and upper bounds. In their BRP model, the bounding constraints must be satisfied. Kadri et al. [35] propose a known threshold levels indicating the minimum and maximum number of bicycles required for each station, and the rebalance demand is used only to meet the boundary of the interval. Different from their research, we define the equilibrium of station as the balance interval, and the inventory of bikes at a station can be beyond this interval. If so, the rebalance utility of the station will decrease. In addition, one objective of our biobjective BRP-BI is to maximize the average rebalance utility of all stations. And another objective is to minimize the vehicle travel distance.

Regarding the model objective and the solution approach, the monoobjective BRPs as well as their corresponding exact or heuristic solution approaches have been studied in many recent works. But the multiobjective counterpart has not widely been considered yet, and only two papers start to go beyond the scope of monoobjective. Mohammadi and Shirouyehzad [36] devise a multiobjective multiperiod BRP to minimize the slacks and surpluses in stations as well as the rebalancing costs. Szeto and Shui [37] investigate a BRP with bilevel objective. The first priority objective is to minimize the total demand dissatisfaction, and the second priority objective is to minimize the service time. However, no multiobjective method has been proposed to solve these problems. For real-life biobjective BRP-

BI, it is more realistic and more useful to calculate the Pareto optimal solutions instead of just one solution. Therefore, we will design a multiobjective solution approach, MS-MOPSO, to solve our biobjective BRP-BI. This will expand the existing research direction of BRP.

3. Mathematical Formulation

BRP-BI studied in this paper is defined on a complete graph $G = (V, A)$. $V = \{0\} \cup N$ denotes the set of all vertices, where $\{0\}$ is the set of depot and $N = \{1, 2, \dots, n\}$ is the set of stations. $A = \{(i, j)\}$ is the set of arcs, and d_{ij} is the distance between vertex i and vertex j .

3.1. Balance Interval. In most literatures, such as the study of Dell'Amico et al. [27], the equilibrium of a BSS station is defined to a prespecified balance point, namely, the target number of bikes in station. Each station must be brought from its current situation to its equilibrium situation to increase the service level of the BSS as well as users' satisfaction. Suppose the equilibrium of station j is b_j and the current inventory of bikes is l_j^1 . If $l_j^1 = b_j$, station j is defined as balanced and its rebalance demand $q_j = 0$. Otherwise, station j is unbalanced, and its rebalance demand $q_j = l_j^1 - b_j$. If $l_j^1 < b_j$, $q_j < 0$, and there is a delivery rebalance demand of supplementary bikes; otherwise, $q_j > 0$, and there is a pickup rebalance demand of surplus bikes, as depicted in Figures 3(a) and 3(b), respectively.

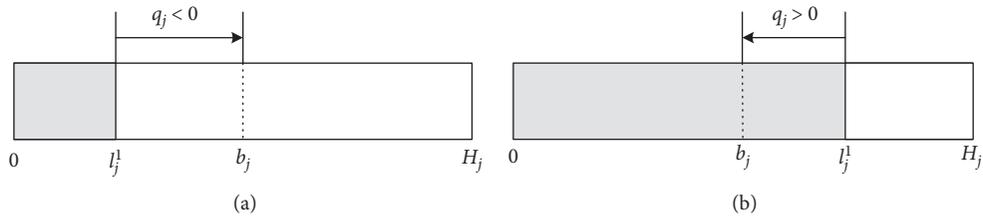
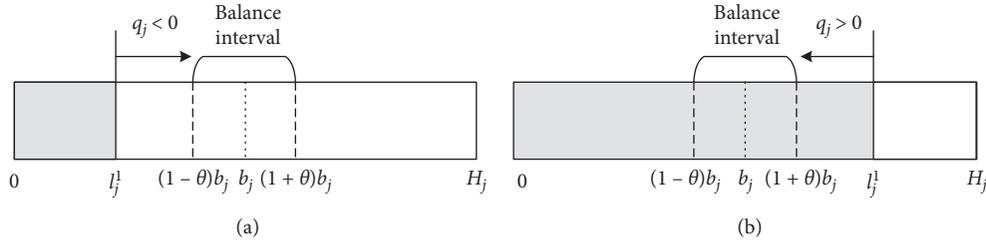
However, from a practical point of view, there are obvious defects. For one thing, due to the capacity limitation of the vehicle, balance point hypothesis can lead to higher rebalance demand of station and longer travel distance. For another, the fluctuation of l_j^1 may increase the number of rebalancing operations.

To avoid the defects caused by the balance point, the equilibrium of station is redefined in this paper as balance interval, that is, an interval of inventory of bikes, depicted in Figure 4. By introducing threshold parameter θ , let $(1 - \theta)b_j$ and $(1 + \theta)b_j$ be the lower and upper bounds of balance interval of station j . Unlike the balance point, if $l_j^1 \in [(1 - \theta)b_j, (1 + \theta)b_j]$, station j is defined as balanced, and its rebalance demand $q_j = 0$. In such a case, station j can be removed from set N , and then $N' = N/\{j\}$ and $V' = \{0\} \cup N'$. If $l_j^1 < (1 - \theta)b_j$, some bikes have to be delivered to station j ; if $l_j^1 > (1 + \theta)b_j$, some bikes need to be picked up from station j . The balance interval can reduce the number of rebalancing operations, thereby reducing the complexity of BRP-BI. Furthermore, the balance point is a special case of our balance interval when $\theta = 0$. Obviously, the different value of θ will result in different sizes of balance interval and then different complexity of BRP-BI. The discussion on the value of θ will be carried out later in Section 5.

3.2. Rebalance Utility. Each station must carry both enough bikes to meet users' rental demands and enough vacant slots to satisfy users' return demands. Since the station has a fixed capacity, a larger quantity of bikes implies a smaller quantity of vacant slots. Therefore, the measurement of station service

TABLE 1: Summary of the literature considering the equilibrium of station.

References	Equilibrium of station	Model objective	Solution approach
Erdoğan et al. [5]	Interval	Single	Benders decomposition, branch-and-cut
Dell'Amico et al. [7]	Point	Single	Branch-and-cut
Cruz et al. [9]	Point	Single	Iterated local search
Duan et al. [10]	Point	Single	Greedy algorithm
Ho and Szeto [20]	Point	Single	Iterated tabu search heuristic
Dell'Amico et al. [27]	Point	Single	L-shaped, branch-and-cut
You [31]	Point	Single	Two-phase heuristic
Kadri et al. [35]	Interval	Single	Branch-and-bound
Mohammadi et al. [36]	Point	Multiple	Lingo
Szeto and Shui [37]	Point	Bilevel	Enhanced artificial bee colony algorithm
This study	Interval	Multiple	MS-MOPSO

FIGURE 3: BSS station with balance point. The grey area represents the current level of occupation, while the white area represents the current vacant slots. H_j is the maximum capacity of station j . (a) Delivery rebalance demand. (b) Pickup rebalance demand.FIGURE 4: BSS station with balance interval. The grey area represents the current level of occupation, while the white area represents the current vacant slots. H_j is the maximum capacity of station j . (a) Delivery rebalance demand. (b) Pickup rebalance demand.

capability includes the bike rental capability as well as the bike return capability, depicted in Figure 5. According to the number of bikes at station j , there are three intervals: $[0, (1 - \theta)b_j]$, $[(1 - \theta)b_j, (1 + \theta)b_j]$, and $[(1 + \theta)b_j, H_j]$.

In the left interval, when the station is empty, bicycles cannot be rented and all slots are available to return, so the rental capability is 0 and the return capability is 1. With the increase in the number of bikes, the rental capability increases, but the return capability is still 1 since there are still enough vacant slots to meet users' return requests. When the number of bikes increases to $(1 - \theta)b_j$, suppose the bikes available are enough for rental, the rental capability becomes 1. Therefore, the station service capability in the left interval depends on the rental capability.

In the middle interval, namely, the balance interval, the station has both enough bikes and enough vacant slots, so the station service capability remains at 1.

In the right interval, as the number of bikes increases, the rental capability remains at 1, but the return capability decreases because there are fewer and fewer vacant slots. When the station is full, on the contrary to empty, the rental capability is still 1 and the return capability is 0. Similarly, the station service capability in the right interval depends on the return capability.

In summary, the station service capability depends on the rental capability in the left interval $[0, (1 - \theta)b_j]$, remains at 1 in the balance interval $[(1 - \theta)b_j, (1 + \theta)b_j]$, and depends on the return capability in the right interval $[(1 + \theta)b_j, H_j]$. In our study, we propose the rebalance utility function to evaluate the service capability of a station after rebalancing operations, whatever the rebalance demand has fully met or not.

Suppose I_j^2 is the inventory of bikes hold by station j after the rebalancing operations, the rebalance utility function $\mu(I_j^2)$ is defined as follows:

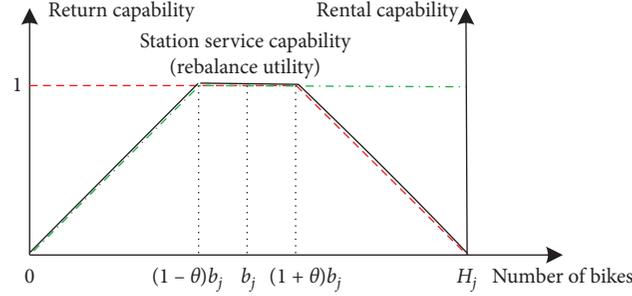


FIGURE 5: Station service capability. Green line represents the rental capability, red line represents the return capability, and black line represents the station service capability (rebalance utility).

$$\mu(l_j^2) = \begin{cases} \frac{l_j^2}{(1-\theta)b_j}, & 0 \leq l_j^2 < (1-\theta)b_j, \\ 1, & (1-\theta)b_j \leq l_j^2 \leq (1+\theta)b_j, \quad \forall j \in N', \\ \frac{H_j - l_j^2}{H_j - (1+\theta)b_j}, & (1+\theta)b_j < l_j^2 \leq H_j, \end{cases} \quad (1)$$

For the purpose of promoting the station service capability and improving user satisfaction, maximizing the average rebalance utility is chosen as one objective of our biobjective BRP-BI.

3.3. Problem Formulations. In BRP-BI, a vehicle with capacity Q is used for the rebalancing operations. It leaves the depot with q^1 inventory of bikes, performs a tour visiting each station exactly once to pick up s_j^+ or deliver s_j^- inventory of bikes at station $j \in N'$ (n' is the number of stations in set N'), and returns to the depot with q^2 inventory of bikes. If the vehicle travels from vertex i to vertex j , the binary decision variable $x_{ij} = 1$; otherwise, $x_{ij} = 0$. Another decision variable f_{ij} represents the quantity of bikes flowing on the arc $(i, j) \in A$. One objective is to maximize the average rebalance utility, and the other one is to minimize the total travel distance.

Referring to the model of Dell'Amico et al. [27], the biobjective mixed-integer programming model for BRP-BI can be written as follows:

$$\min g_1 = \sum_{i,j \in V', i \neq j} x_{ij} d_{ij}, \quad (2)$$

$$\max g_2 = \left(\frac{1}{n'}\right) \sum_{j \in N'} \mu(l_j^2), \quad (3)$$

s.t.

$$\sum_{i \in V', i \neq j} x_{ij} = 1, \quad \forall j \in V', \quad (4)$$

$$\sum_{i \in V', i \neq j} x_{ij} - \sum_{i \in V', i \neq j} x_{ji} = 0, \quad \forall j \in N', \quad (5)$$

$$\sum_{i \in S} \sum_{i \in S} x_{ij} \leq |S| - 1, \quad \forall S \subseteq V', 2 \leq |S| \leq n' - 1, \quad (6)$$

$$0 \leq f_{ij} \leq Qx_{ij}, \quad \forall (i, j) \in A, \quad (7)$$

$$\sum_{i \in V', i \neq j} f_{ji} - \sum_{i \in V', i \neq j} f_{ij} = s_j^- - s_j^+, \quad \forall j \in N', \quad (8)$$

$$l_j^2 - l_j^1 = s_j^- - s_j^+, \quad \forall j \in N', \quad (9)$$

$$s_j^+ \times s_j^- = 0, \quad \forall j \in N', \quad (10)$$

$$0 \leq s_j^+ \leq Q - \sum_{i \in V', i \neq j} f_{ij}, \quad \forall j \in N', \quad (11)$$

$$0 \leq s_j^- \leq \sum_{i \in V', i \neq j} f_{ij}, \quad \forall j \in N', \quad (12)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V', i \neq j. \quad (13)$$

Objective function (2) minimizes the total travel distance, and objective function (3) maximizes the average rebalance utility. Obviously, these two objectives cannot be optimized at the same time. Equation (4) imposes that each station has to be visited exactly once. Equation (5) ensures flow conservation, and constraint (6) eliminates subtours. Constraint (7) guarantees that the vehicle capacity is not

exceeded throughout its tour. Constraints (8) and (9) specify that the difference between the flows entering and leaving a station is equal to its rebalance demand. Constraint (10) enforces that a station has either delivery rebalance demand or pickup rebalance demand and cannot have both. Constraint (11) guarantees that the bikes picked up at a station cannot exceed the remaining capacity of the vehicle. Constraint (12) enforces that the bikes delivered to a station cannot exceed the inventory hold by the vehicle. Furthermore, constraint (13) restricts the decision variables that are binary.

4. Solution Method

In recent years, many efficient metaheuristics have been proposed to solve various multiobjective optimization problems, such as EM-BBO [38], NSGA-II [39], MOEA/D [40], and MOPSO [41]. Compared with single-objective methods, multiobjective algorithms can generate a set of nondominated optimal solutions, from which suitable schemes can be selected for different actual scenarios. Due to the advantages of simplicity, low-cost implementation, and fast convergence, MOPSO has become a well-known algorithm for optimizing practical multiobjective problems. To address our biobjective BRP-BI, MS-MOPSO, an improved version of the MOPSO proposed by Coello et al. [42], is presented in this section. In order to achieve balanced exploration and exploitation capabilities, we have introduced variable neighborhood search [43] and multistart strategy [44] to enhance our method. The former can avoid falling into local optima and expand exploration capabilities. The latter can help find higher quality Pareto optimal solutions. The procedure and key components of our algorithm are described in detail below.

4.1. Procedure of the Algorithm. The procedure of our MS-MOPSO is demonstrated in Figure 6. Unlike single-objective PSO, there are two different criteria to determine the personal optimum $pbest$ and the global optimum $gbest$. We first select the unbalanced stations from the BSS and then use the fitness evaluation function to obtain the nondominated particles. All the nondominated particles are gathered into an external repository (REP), from which every particle chooses its $gbest$. For $pbest$, domination-based and probabilistic rules are utilized. To avoid being trapped in local optima, we have introduced the crossover operator of genetic algorithm to update the population and the variable neighborhood search to improve the exploratory capability of our algorithm. Finally, when the number of runs for multistart strategy (ms) reaches the specified number, we will get the nondominated optimal solutions for biobjective BRP-BI.

4.2. Initial Solutions Generation. The stations with inventory of bikes in their balance intervals can be regarded as in equilibrium and do not need to be rebalanced. For example, there are 15 stations, numbered 1–15, whose current inventories of bikes are illustrated in Table 2.

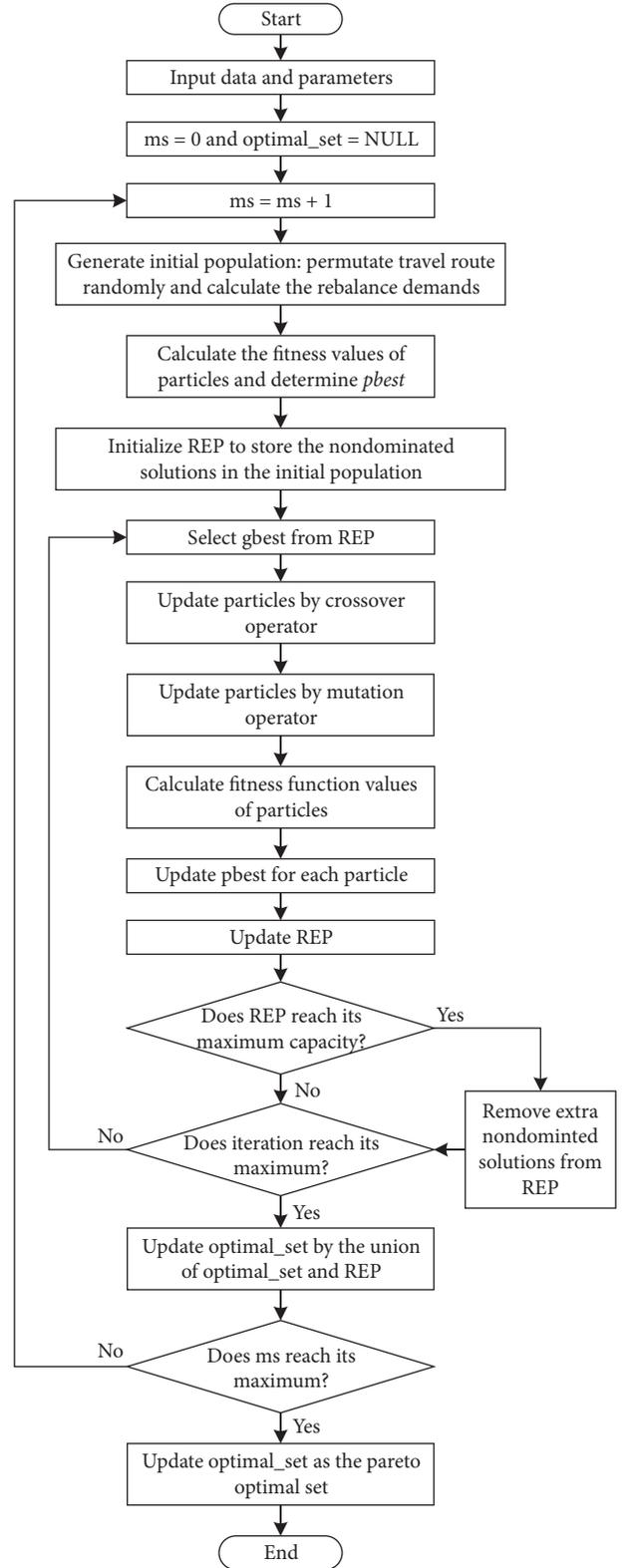


FIGURE 6: Procedure of the MS-MOPSO.

Suppose the target inventory of stations is 50 and $\theta = 0.2$, the balance interval is $[40, 60]$. It can be seen from Table 2 that the current inventories of bikes in stations 2, 3, 8, 9, and 14 are in the balance interval. Thus, these balanced stations

TABLE 2: The inventories of stations.

Stations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Current inventories	98	55	48	32	19	87	27	54	55	27	97	26	31	46	38

The italic stations are balanced.

can be removed from the BSS and will not be visited in the rebalancing operations. In brief, the vehicle only needs to visit the unbalanced stations, and this will be helpful to reduce the scale of the problem as well as the total travel distance.

A feasible solution of the BRP-BI is made up of two parts: one is the vehicle route and the other is the rebalance demand. In the following, we will discuss how to construct the initial vehicle route and how to calculate the rebalance demand for each unbalanced station.

4.2.1. Vehicle Route Construction. A vehicle route can be represented as $(0, i_1, i_2, \dots, i_n, 0)$, where 0 is the depot and i_1, i_2, \dots, i_n is a sequence of unbalanced stations that need to be rebalanced. By random permutation for i_1, i_2, \dots, i_n , a typical “vehicle route” is displayed in the first line of Table 3, where depot 0, as the start and end points, is omitted. Another vehicle route can surely be obtained by changing the permutation of stations. In other words, different permutations of stations correspond to different vehicle routes.

4.2.2. Rebalance Demand Calculation. The notations used in this section are set as follows: inv_{i1} and inv_{i2} are the inventories in station i before and after rebalancing operations, respectively; $ideal_inv$ is the target inventory; lb/ub is the lower/upper boundary of the balance interval; q_i is the load on the vehicle after visiting station i ; and Q is the capacity of vehicle. The pseudocode for calculation of rebalance demand of station i ($i = 1, 2, \dots, n$) is demonstrated in Algorithm 1. As mentioned above, the balanced stations have already been eliminated from the rebalancing operations, thus inv_{i1} is either less than lb or more than ub .

If $inv_{i1} < lb$, there is a delivery rebalance demand at station i , and the unloading quantities depend on the current load on vehicle. If the load is enough, station i will be replenished to the target inventory $ideal_inv$, which is commonly in the middle of the balance interval. Otherwise, the load is insufficient to satisfy the delivery demand, and station i will get all the load on vehicle. In such a case, if $inv_{i2} < lb$, the rebalance utility of station i will decrease.

On the contrary, if $inv_{i1} > ub$, there is a pickup rebalance demand at station i , and the loading quantities depend on the free space of vehicle. If the free space is enough, the extra inventory will be loaded into the vehicle to make the inventory of station i be $ideal_inv$; otherwise, the vehicle will be filled. Similarly, if $inv_{i2} > ub$, its rebalance utility will also decrease.

In Table 3, the “inventories before rebalancing” of the unbalanced stations also comes from Table 2. Given the target inventory $ideal_inv = 50$, the “rebalance demands” can be obtained by “inventories before rebalancing” minus

$ideal_inv$, where “+” means the pickup rebalance demand and “−” means the delivery rebalance demand.

Suppose the vehicle’s capacity is 50 and its initial load is zero, the “vehicle’s load after rebalancing” and “inventories after rebalancing” have to be calculated iteratively by using Algorithm 1. In the first two stations, the rebalancing operations perform well, and the vehicle picks up 48 bikes at station 1 and unloads 18 bikes at station 2. When the vehicle arrives at station 5, its load is 30 bikes, but the delivery demand of station 5 is 31 bikes. According to Algorithm 1, station 5 is replenished by only 30 bikes and is not fully satisfied. This can avoid increasing the travel distance because the vehicle must return back to the depot to pick up one more bike if fully meeting the demands is required. In other words, we have trade-off between the travel distance and the rebalance utility. The similar analysis can be applied to the remaining stations.

Moreover, the “quantities satisfied” are the differences between “inventories before rebalancing” and “inventories after rebalancing.” By comparing the “rebalance demands” and “quantities satisfied,” the grey columns demonstrate that station 5, 10, and 15 have insufficient bikes and they are not fully satisfied. However, the rebalancing operations are satisfying because “inventories after rebalancing” for all stations is in the balance interval $[40, 60]$. This also illustrates the superiority of our balance interval.

In summary, to generate the initial solution population, different vehicle routes can be constructed by random permutation, and the rebalance demands can be calculated by using Algorithm 1.

4.3. Fitness Function and Personal Best Particle. Particles in PSO are sharing information and moving towards both their own personal best particle $pbest$ and the global best particle $gbest$. But in multiobjective optimization, “nondominated” takes the place of “best.” And there is more than one criterion to determine the nondominated particles in our MS-MOPSO. In this section, we will discuss how to choose the $pbest$ and then in the following, $gbest$ will be involved.

In the biobjective BRP-BI, the dominance relationship of the particles is obtained by comparing their fitness functions originated from the objective functions (2) and (3). The fitness functions used in our algorithm are specified as follows:

$$f_1 = \sum_{i,j \in V', i \neq j} x_{ij} d_{ij}, \quad (14)$$

$$f_2 = -\left(\frac{1}{n}\right) \sum_{j \in N'} \mu(l_j^2). \quad (15)$$

Based on these fitness functions, the Pareto dominance in our algorithm is defined as follows: for two particles $z1$

TABLE 3: An example of the initial solution.

Vehicle route	1	4	5	6	7	10	11	12	13	15
Inventories before rebalancing	98	32	19	87	27	27	97	26	31	38
Rebalance demands	+48	-18	-31	+37	-23	-23	+47	-24	-19	-12
Vehicle's load after rebalancing	48	30	0	37	14	0	47	23	4	0
Inventories after rebalancing	50	50	49	50	50	41	50	50	50	42
Quantities satisfied	+48	-18	-30	+37	-23	-14	+47	-24	-19	-4

The stations in italics demonstrate that their demands are not fully met.

```

If  $inv_{i1} < lb$ 
  if  $ideal\_inv - inv_{i1} \leq q_{i-1}$ 
     $q_i = q_{i-1} - (ideal\_inv - inv_{i1})$ 
     $inv_{i2} = ideal\_inv$ 
  else
     $q_i = 0$ 
     $inv_{i2} = inv_{i1} + q_{i-1}$ 
  end
else
  if  $inv_{i1} - ideal\_inv \leq Q - q_{i-1}$ 
     $q_i = q_{i-1} + (inv_{i1} - ideal\_inv)$ 
     $inv_{i2} = ideal\_inv$ 
  else
     $q_i = Q$ 
     $inv_{i2} = inv_{i1} - (Q - q_{i-1})$ 
  end
end
end

```

ALGORITHM 1: Rebalance demand calculation.

and z_2 , if $f_1(z_1) \leq f_1(z_2)$ and $f_2(z_1) < f_2(z_2)$, or $f_1(z_1) < f_1(z_2)$ and $f_2(z_1) \leq f_2(z_2)$, z_1 dominates z_2 . Similarly, if $f_1(z_2) \leq f_1(z_1)$ and $f_2(z_2) < f_2(z_1)$, or $f_1(z_2) < f_1(z_1)$ and $f_2(z_2) \leq f_2(z_1)$, z_2 dominates z_1 . Otherwise, there is no dominance relationship between z_1 and z_2 .

Particle z^* is said to be a nondominated particle, or Pareto particle, if and only if $f_1(z^*) \leq f_1(z)$ and $f_2(z^*) < f_2(z)$, or $f_1(z^*) < f_1(z)$ and $f_2(z^*) \leq f_2(z)$ for any particle z . If z^* is a nondominated particle, $(f_1(z^*), f_2(z^*))$ is named as a nondominated point and all these points constitute the Pareto front.

In an initial solution, for each particle, $pbest$ is itself. In the process of solving, if the current particle can dominate its $pbest$, $pbest$ is updated to this particle; if $pbest$ dominates the current particle, $pbest$ is not changed; otherwise, the current particle and $pbest$ have no dominance relationship, and one of them is randomly selected as $pbest$.

4.4. REP and Global Best Particle. In our MS-MOPSO, REP is used to keep a historical record of all nondominated solutions found during the search process. At the beginning of the search, the nondominated particles in the initial solution are all added to REP. Along the search process, the current nondominated particles found at each iteration are compared one by one with all the solutions in REP. If one current particle is dominated by an individual of REP, it is discarded. Otherwise, if all individuals of REP cannot dominate this

current particle, such particle can be appended to REP. If there are solutions of REP dominated by the new member, those individuals are removed from REP. If REP has reached its maximum capacity, we use the adaptive grid method proposed by Coello et al. [42] to remove extra solutions and keep the solutions in REP well distributed.

For the selection of $gbest$ from REP, we refer to the method of Dabiri et al. [41]. The solution space is divided into multiple equal grids, and the grids with fewer particles have a higher chance of being selected. If there is more than one particle in the grid selected, one of them is taken randomly as $gbest$.

4.5. Particle Crossover Operator. The traditional position and velocity update formulas cannot directly apply to renew the particles in MS-MOPSO. Therefore, the crossover operator of the genetic algorithm is introduced to update the particles. And our particle crossover operator is defined as follows:

$$X_i(t+1) = X_i(t) \otimes pbest_i(t) \otimes gbest(t), \quad (16)$$

where $X_i(t+1)$ and $X_i(t)$ are the solutions of the i^{th} particle in the current iteration (t) and the next iteration ($t+1$), respectively; $pbest_i(t)$ is the personal optimal solution of the i^{th} particle, and $gbest(t)$ is the global optimal solution; and symbol \otimes is on behalf of the crossover operator.

An example is taken to illustrate the framework of our crossover operator. As shown in Figure 7(a), $X_i(t)$ and its $pbest_i$ are initially selected as the parents P_1 and P_2 , and then the vehicle routes are updated by exchanging their segments of chromosome selected randomly and the corresponding rebalance demands are computed using Algorithm 1. After crossover operations, two offspring particles (O_1 and O_2) are generated, displayed in Figure 7(b). They are compared by their fitness function values computed by formulas (14) and (15), and the nondominated one is selected to perform the subsequent operations. If there is no dominance relationship, one of them is randomly selected. In this example, O_2 is selected as a new parent P'_1 and $gbest$ is the other parent P'_2 . In the same way, P'_1 and P'_2 are compared to produce their offsprings O'_1 and O'_2 , demonstrated in Figure 7(c). Similarly, we take the nondominated individual or a random one (if there is no dominance relationship) as the updated particle $X_i(t+1)$.

4.6. Particle Mutation Operator. For the purpose of increasing the diversity of particles, expanding the search range, and jumping out of local optimum, the particle mutation operator enhanced by variable neighborhood

$X_i(t)(P_1)$	0	11	7	12	15	6	5	13	10	4	1	0
$pbest_i(P_2)$	0	7	5	12	1	11	15	13	10	4	6	0
(a)												
O_1	0	<u>5</u>	7	12	1	11	15	13	10	4	<u>6</u>	0
$O_2(P'_1)$	0	7	<u>1</u>	12	15	6	5	13	10	4	<u>11</u>	0
$gbest(P'_2)$	0	11	4	13	6	1	15	5	12	10	7	0
(b)												
O'_1	0	11	4	13	6	<u>1</u>	5	<u>7</u>	10	<u>12</u>	<u>15</u>	0
$O'_2(X_i(t+1))$	0	7	1	12	15	<u>4</u>	<u>6</u>	5	<u>11</u>	10	<u>13</u>	0
(c)												

FIGURE 7: An example for particle crossover operator. The grey segments are the chromosomes selected to cross.

search is proposed to update the particles. There are three neighborhood structures used in our algorithm:

NS1: one station is selected randomly and removed from its original position, and then it is inserted into another random position, depicted in Figure 8(a).

NS2: two stations are selected randomly, and their positions are swapped, displayed in Figure 8(b).

NS3: two stations are selected randomly, and the sequence between them is arranged in the reverse order, shown in Figure 8(c).

In our mutation operator, the three structures above are performed randomly. After the mutation operations, if the new particle dominates the original, the new particle will replace the original; if the original dominates the new particle, the new particle will be discarded. Otherwise, if there is no dominance relationship between these two particles, then one of them is randomly chosen.

5. Results and Discussion

We have implemented the MS-MOPSO algorithm using MATLAB R2014b, and all computational experiments have been carried out on a notebook with Intel Core i5-8265U CPU and 8 GB RAM. The test instances for numerical analysis are generated randomly.

5.1. Test Instances and Parameters. The performance of MS-MOPSO is evaluated by 12 test instances (3 types, 4 instances per type), displayed in Table 4. These three types are small size ($n = 20$), medium size ($n = 60$), and large size ($n = 100$). In these instances, the stations' coordinates are randomly generated in the range of longitude $31.15^\circ \sim 31.25^\circ$ and latitude $121.33^\circ \sim 121.43^\circ$. The depot was assumed to be located at the center of this area, and its coordinate is $(31.20^\circ, 121.38^\circ)$. The station capacity is 100, and its current quantity of bikes is randomly drawn from a range $[0, 100]$. A single vehicle with capacity of 50 is used to serve the unbalanced stations selected in advance. Unless otherwise specified, we set $\theta = 0.2$ in all instances; thus, the balance interval is $[40, 60]$. The numbers of unbalanced stations are also shown in Table 4.

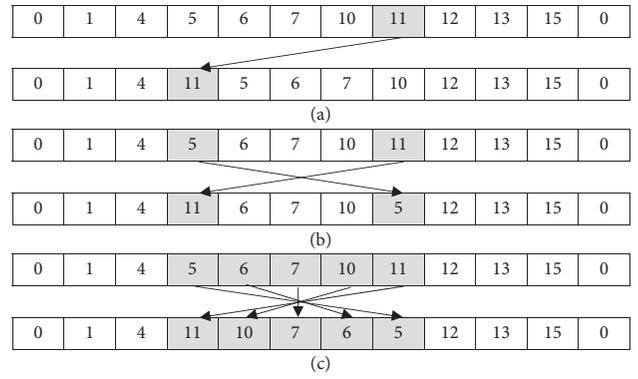


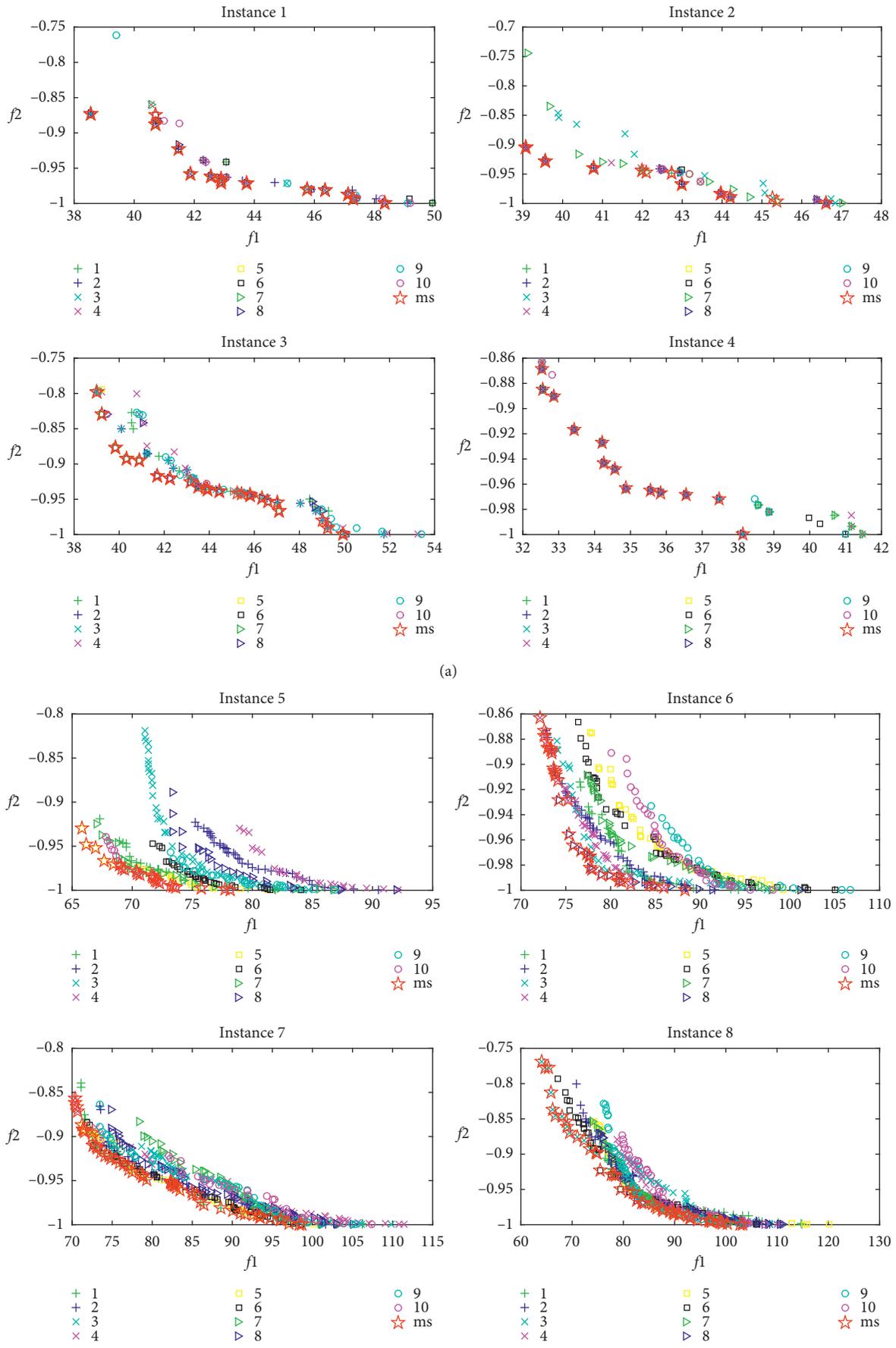
FIGURE 8: Neighborhood structures. The grey stations change their positions in the original route. (a) NS1. (b) NS2. (c) NS3.

TABLE 4: Test instances. 12 instances of 3 types.

Instances	Small size				Medium size				Large size			
	1	2	3	4	5	6	7	8	9	10	11	12
n	20	20	20	20	60	60	60	60	100	100	100	100
Unbalanced stations	15	16	16	15	46	48	43	43	75	87	78	85

In the following section, the number of runs for multistart strategy is 3 for small-size instances, 4 for medium-size instances, and 5 for large-size instances. The parameters of MS-MOPSO and MOPSO are set as follows: particle population is 100, REP size is 100, and mutation rate is 1. And HNSGA-II runs using a population size of 100, a crossover rate of 0.7, a mutation rate of 1, and a mutation percentage of 0.4. For the small-size, medium-size, and large-size instances, the maximal iterations are set to 300, 800, and 1800, respectively.

5.2. Advantage of Multistart Strategy. To reveal the benefit of multistart strategy, we perform our algorithm without multistart strategy for 10 times and get 10 Pareto optimal solution sets for each instance, numbered $1, 2, \dots, 10$, depicted in Figure 9. And then we compare them with the Pareto optimal solution set ("ms" in Figure 9) obtained by our MS-MOPSO



(a)

(b)

FIGURE 9: Continued.

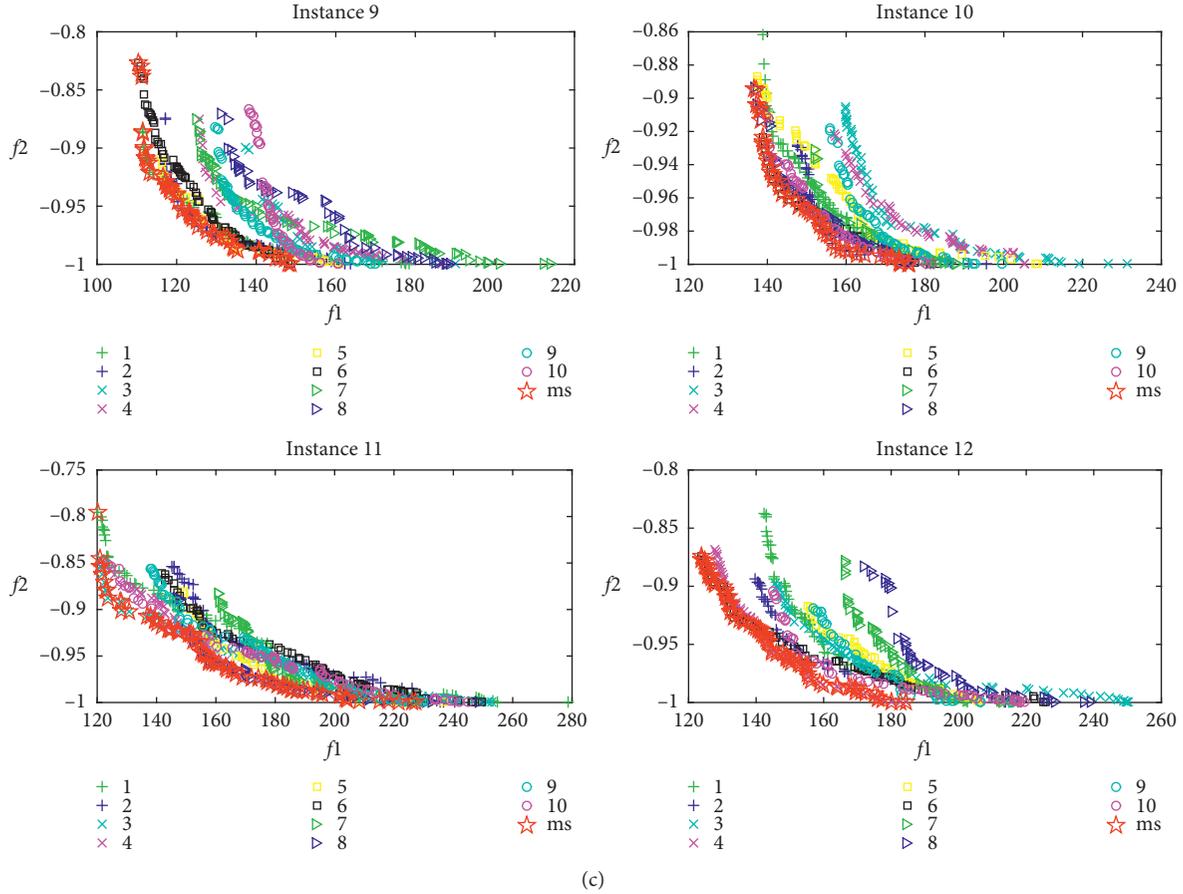


FIGURE 9: Computation results with or without multistart strategy. (a) Small-size instances. (b) Medium-size instances. (c) Large-size instances.

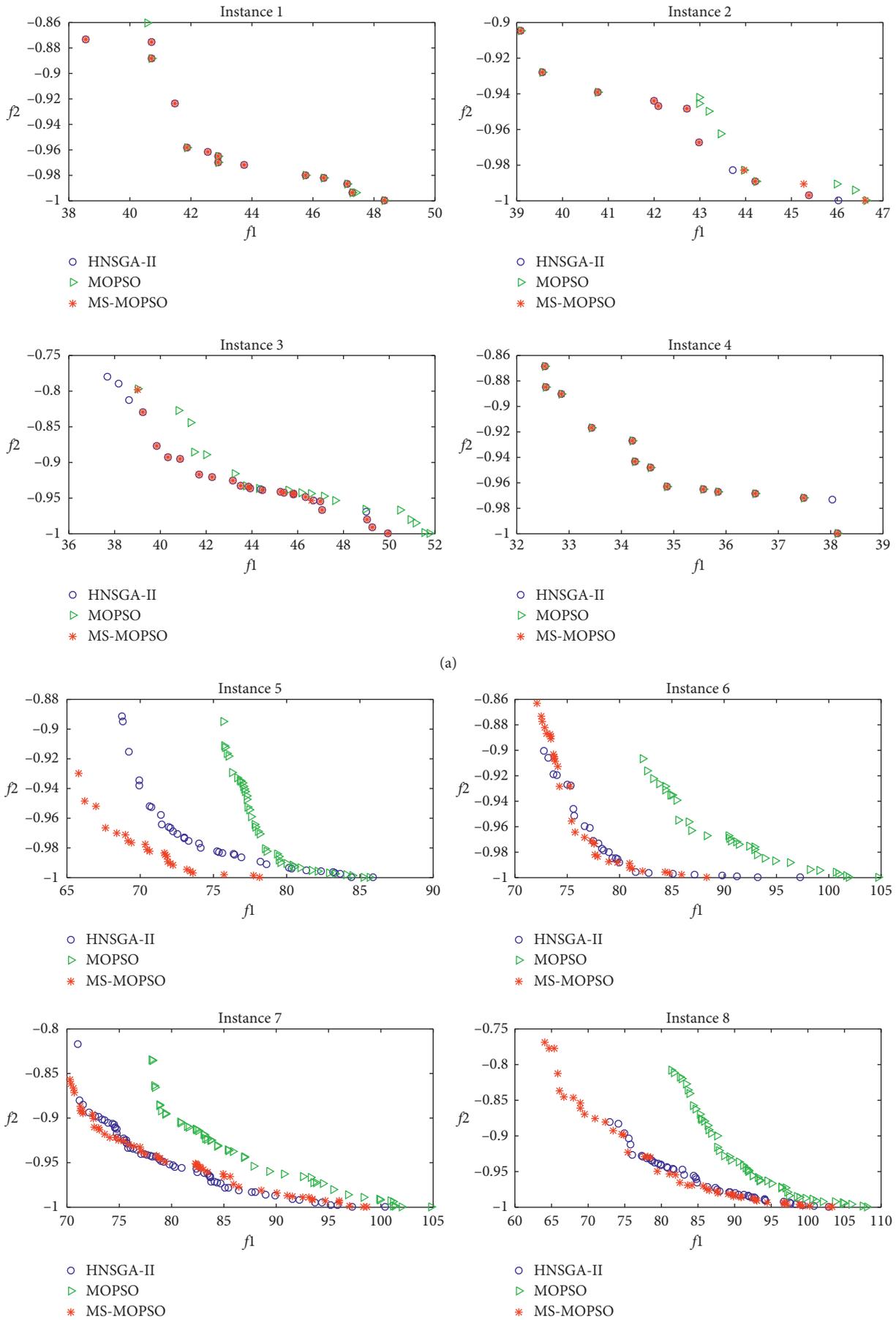
using multistart strategy. The objective $f1$ is the total travel distance (formula (14)), and the other objective $f2$ is the minus average rebalance utility (formula (15)). The comparison results in Figure 9 demonstrate that the “ms” solution set is significantly better than the others, and the advantage is particularly obvious in the medium-size and large-size instances. Therefore, multistart strategy of our algorithm is conducive to finding the better Pareto optimal solutions.

5.3. Comparisons with HNSGA-II and MOPSO. To validate the efficiency and efficacy of our algorithm, we compare it with HNSGA-II and MOPSO, using two criteria: the quality of solutions and the running time.

First, we compare the three algorithms by the quality of solutions. Figure 10 displays the graphical results of the Pareto optimal solutions in all instances. For the small-size instances, it is noticeable that the difference is not large. The Pareto optimal solutions for instances 1 and 4 are almost the same. But in instances 2 and 3, the quality of solutions obtained by MS-MOPSO and HNSGA-II is still slightly better than that obtained by MOPSO. For the medium-size and large-size instances, the performance of MS-MOPSO is obviously better than that of MOPSO in all instances. In

instances 6, 7, and 11, MS-MOPSO and HNSGA-II have similar performance. But in instances 5, 8, 9, 10, and 12, MS-MOPSO performs better. Especially in instances 5 and 10, it achieves complete superiority. Though some solutions produced by MS-MOPSO are slightly worse than HNSGA-II, the performance of our MS-MOPSO is satisfying if we consider that HNSGA-II is normally considered good enough to solve most multiobjective optimization problems.

Then we compare the three algorithms by running time. We run 10 times for each instance and each algorithm, and the average running time is shown in Figure 11. Although the solutions of MOPSO are not the best, it has complete advantages in running time in all instances. Obviously, MS-MOPSO takes longer than MOPSO due to the multistart strategy. However, in the small-size instances, MS-MOPSO has a shorter running time than HNSGA-II. In the medium-size instances, the running time of MS-MOPSO is slightly longer than that of HNSGA-II. In the large-size instances, MS-MOPSO requires nearly 3000 seconds, which is also acceptable to actual BSS operators because the application scenario of BRP-BI is at night and there is enough time to calculate a better solution. Furthermore, our experiments are conducted on a notebook. For a more powerful computer, the running time of MS-MOPSO will be greatly reduced.



(b)

FIGURE 10: Continued.

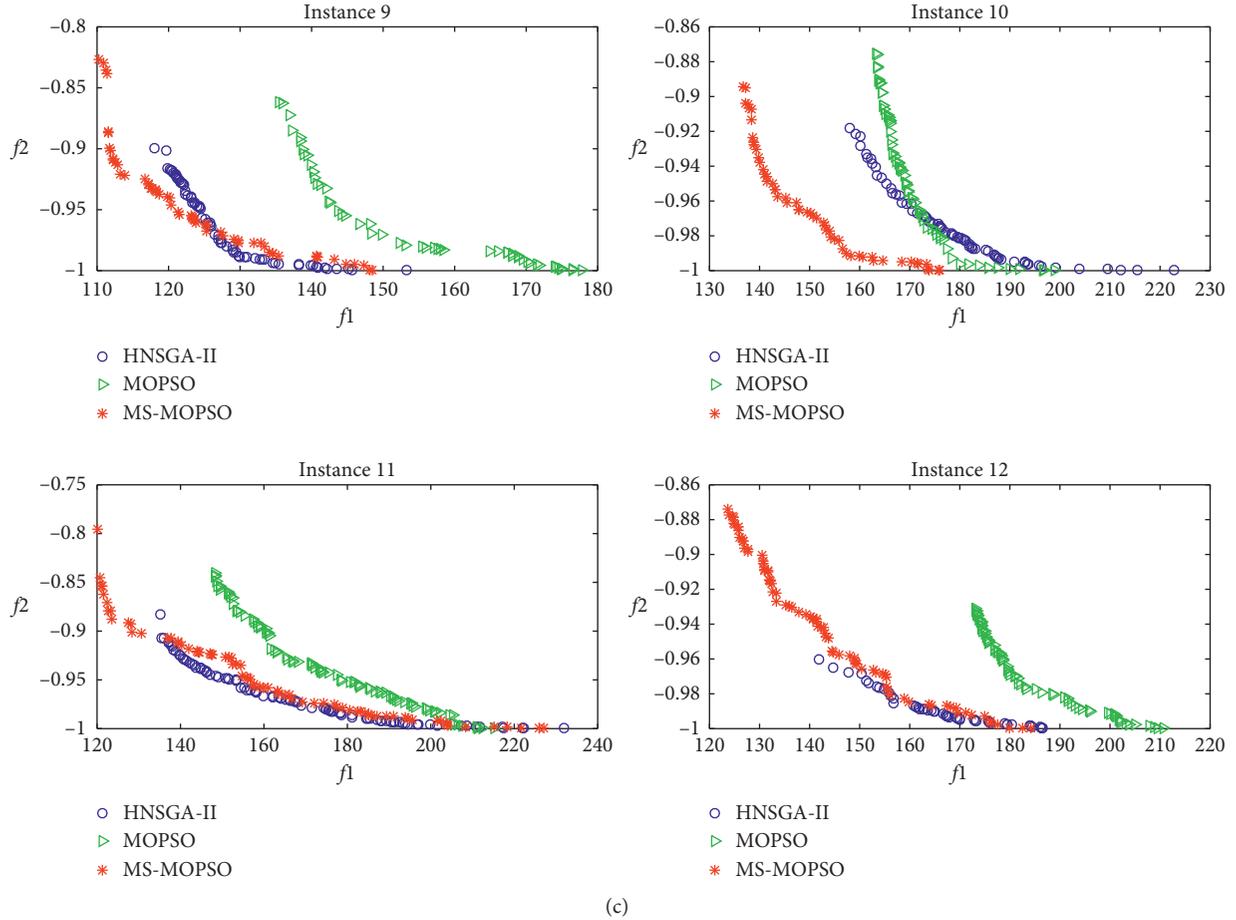


FIGURE 10: Pareto optimal solutions produced by MS-MOPSO, HNSGA-II, and MOPSO. (a) Small-size instances. (b) Medium-size instances. (c) Large-size instances.

5.4. Statistical Analysis of Performance Metrics. To evaluate the convergence and the diversity of multiobjective algorithms, several performance metrics are proposed, such as hypervolume (HV), inverse generation distance (IGD), and spacing metrics (SM) [45, 46]. HV is a comprehensive index used to evaluate both convergence and diversity. The bigger the HV is, the better the nondominated solution set is. IGD is the average distance between the Pareto optimal front and the nondominated solution set obtained by the algorithm. The smaller the IGD is, the better the diversity and the convergence of the nondominated solution set are. SM measures how well distributed the nondominated solutions are. The smaller the SM is, the better the solutions spread along the obtained front are.

All the metrics above require a true Pareto front. Unfortunately, the true Pareto front of our biobjective BRP-BI is unknown since it is impossible to be exactly solved. To overcome this drawback, Baños et al. [45] propose a way to construct a quasi-optimal Pareto front by executing all algorithms over a very large number of iterations. After that, all the nondominated solutions are combined, discarding the dominated solutions and maintaining the nondominated ones. In our study, the same method is used to construct the quasi-optimal Pareto front for our biobjective BRP-BI. Each

algorithm has been executed for 3 hours for each instance, and the composed quasi-optimal Pareto front is used to evaluate the performance of multiobjective algorithms.

To provide reliable statistics, each algorithm is run 10 times for instance 4 (small size), instance 8 (medium size), and instance 12 (large size). To calculate HV, we set the reference point as twice the objective f_1 and half the objective f_2 , where f_1 and f_2 are the average value of objective values in the quasi-optimal Pareto front. The statistical results of HV, IGD, and SM obtained by using MS-MOPSO, HNSGA-II, and MOPSO are displayed in Tables 5–7, respectively. In all the three tables, the best, worst, average, and variance of the three metrics are given. The best values are marked with bold fonts.

It is observed that although the best and worst values of HV obtained by MS-MOPSO are equal to those obtained by HNSGA-II in instance 4, the average HV obtained by MS-MOPSO are higher in all instances. This means that regardless of the size of the BRP-BI, MS-MOPSO outperforms HNSGA-II and MOPSO in terms of convergence and diversity. Furthermore, the comparison results of the average values of IGD also confirm this conclusion. With respect to the average values of SM, the values corresponding to the solutions obtained by MS-MOPSO are also superior to those obtained by

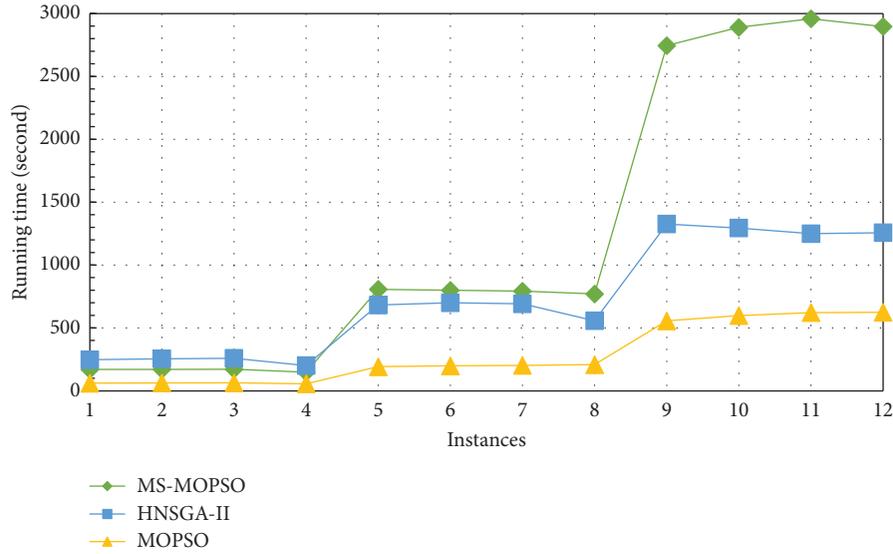


FIGURE 11: The running times for 12 instances.

TABLE 5: Results of HV for three instances.

		MS-MOPSO	HNSGA-II	MOPSO
Instance 4	Best	19.58	19.58	19.51
	Worst	19.52	19.52	16.74
	Average	19.57	19.55	18.41
	Variance	0.0002	0.0005	1.2424
Instance 8	Best	46.81	43.11	37.85
	Worst	44.10	36.85	29.02
	Average	45.7462	40.3282	33.71313
	Variance	0.8416	3.4812	7.1756
Instance 12	Best	79.27	71.39	54.83
	Worst	67.35	54.34	40.83
	Average	72.83	62.61	48.71
	Variance	8.8486	19.2971	24.5173

TABLE 6: Results of IGD for three instances.

		MS-MOPSO	HNSGA-II	MOPSO
Instance 4	Best	0.007	0.00005	0.0194
	Worst	0.019	0.0257	0.1974
	Average	0.0081	0.0154	0.0874
	Variance	1.190E-05	6.793E-05	4.533E-03
Instance 8	Best	0.0054	0.0204	0.0505
	Worst	0.0146	0.0523	0.1018
	Average	0.0093	0.0338	0.0738
	Variance	8.860E-06	8.810E-05	2.609E-04
Instance 12	Best	0.0127	0.0522	0.1183
	Worst	0.0538	0.1202	0.1981
	Average	0.0301	0.0799	0.1533
	Variance	0.0001	0.0003	0.0008

the other two algorithms. This means that the nondominated solutions obtained by MS-MOPSO are well distributed.

In addition, the statistical results of HV and IGD show that the variance of MS-MOPSO is the smallest in all

instances. Although the variance of SM is not the best, it is noticeable that if HV and IGD are not good, the comparison of SM is meaningless because this metric is used only to measure the distribution of nondominated solutions in the

TABLE 7: Results of SM for three instances.

		MS-MOPSO	HNSGA-II	MOPSO
Instance 4	Best	0.3216	0.3292	0.3648
	Worst	0.531	0.6136	0.7863
	Average	0.3406	0.4587	0.6078
	Variance	0.0036	0.0125	0.0195
Instance 8	Best	0.5183	0.7076	0.7465
	Worst	0.7152	0.9365	0.885
	Average	0.6362	0.8263	0.8043
	Variance	0.0037	0.0054	0.0023
Instance 12	Best	0.6876	0.8462	0.8204
	Worst	0.8761	1.024	0.9294
	Average	0.7485	0.9169	0.8628
	Variance	0.0026	0.0021	0.0013

TABLE 8: Sensitivity analysis results of different θ .

θ	Balance interval	Unbalanced stations	Unbalanced ratio (%)	$g_1(f_1)$	$g_2(-f_2)$
0.3	[35, 65]	38	63.3	75.5880	0.9286
0.28	[36, 64]	39	65.0	76.1066	0.9245
0.26	[37, 63]	40	66.7	78.6623	0.9462
0.24	[38, 62]	40	66.7	80.2704	0.9365
0.22	[39, 61]	42	70.0	80.6227	0.9439
0.2	[40, 60]	43	71.7	79.1583	0.9325
0.18	[41, 59]	44	73.3	89.7473	0.9510
0.16	[42, 58]	46	76.7	89.9612	0.9504
0.14	[43, 57]	47	78.3	93.0851	0.9441
0.12	[44, 56]	49	81.7	95.5070	0.9518
0.1	[45, 55]	53	88.3	100.3936	0.9540
0.08	[46, 54]	53	88.3	101.5975	0.9486
0.06	[47, 53]	54	90.0	102.0251	0.9447
0.04	[48, 52]	55	91.7	106.2414	0.9509
0.02	[49, 51]	58	96.6	113.9910	0.9559
0	50	59	98.3	117.2586	0.9490

objective space. Therefore, one conclusion can be drawn: our MS-MOPSO is stable since the smaller the variance, the smaller the fluctuation.

To sum up, compared with HNSGA-II and MOPSO, MS-MOPSO performs better in convergence, diversity, and distribution. Besides, MS-MOPSO is more stable. MS-MOPSO is an improved version of MOPSO and keeps the advantages of simplicity, diversity, and fast convergence. By combining the multistart strategy as well as variable neighborhood search to enhance its exploratory capability, MS-MOPSO has achieved a well-balanced exploration-exploitation capability. Therefore, our algorithm can generate higher quality Pareto optimal solutions than HNSGA-II and MOPSO in the biobjective BRP-BI.

5.5. Discussion on the Value of θ . Parameter θ directly affects the size of the balance interval and then the number of unbalanced stations that needs to be served. In other words, the change of θ will result in the change of the balance interval, which can affect whether the stations need to be rebalanced or not. Generally speaking, the larger θ is, the less the unbalanced stations are. In this section, the sensitivity analysis of θ is

analyzed using the instance 8, a medium-size instance of 60 stations. Different values of θ are from 0.3 to 0 in decrement of 0.02.

Table 8 displays the balance interval, the number of unbalanced stations, the unbalanced ratio, and two objectives for different θ . With the decrease of θ , the balance interval is getting narrower and narrower. Thus, both the number of unbalanced stations and unbalanced ratio are larger and larger. The objective $g_1(f_1)$, i.e., the total travel distance, generally increases with the decrease of θ . Especially when $\theta = 0$, the total travel distance is the highest because the balance interval shrinks to a point; that is, the station is in equilibrium only when its inventory is just 50. Obviously, there is a fluctuation. When the value of θ is reduced from 0.22 to 0.2, the number of unbalanced stations increases from 42 to 43, but the value of $g_1(f_1)$ decreases from 80.6227 to 79.1583. This is because that the biobjective BRP-BI has both pickup demand and delivery demand, so the sequence of stations visited by the vehicle will affect the travel distance. Due to the fixed capacity of the vehicle, when the load is almost full, the vehicle may have to visit a remote delivery station. If the added station is near the current station and has a delivery demand, the travel distance can be

reduced. Conversely, when the load is almost empty, there may exist a station with pickup demand, which can also shorten the travel distance. As for the objective $g_2(-f_2)$, with the decrease of θ , the average rebalance utility also keeps the uptrend with tiny fluctuations.

It can be observed that $\theta = 0.2$ may be an appropriate value for instance 8. For other instances, the sensitivity analysis can be performed by solving a series of biobjective BRP-BI problems with different θ . According to the computation results, the suitable value of θ can be determined. However, this method cannot be widely used in real world since it is time-consuming. For the real-life BSS operators, a more realistic method is that they can use big data methods to find the appropriate value of θ from historical data.

6. Conclusions

We have introduced, modeled, and solved the biobjective BRP-BI, a multiobjective variant of BRP encountered in the BSS operations. We first extend the station's equilibrium from balance point to balance interval and then formulate the BRP-BI as a biobjective mixed-integer programming with the objectives of minimizing the total travel distance as well as maximizing the average rebalance utility. Furthermore, the MS-MOPSO algorithm is enhanced by the crossover operator, and variable neighborhood search is proposed to solve this problem. To illustrate the efficiency and efficacy of our algorithm, it has been tested on three types of twelve instances. The computational results reveal the huge advantage of multistart strategy and MS-MOPSO. The statistical analysis results of three performance metrics (i.e., hypervolume, inverse generation distance, and spacing metrics) also confirm the same conclusion: MS-MOPSO achieves better nondominated optimal solutions in terms of convergence, diversity, and distribution, outperforming those obtained by HNSGA-II and MOPSO. Furthermore, MS-MOPSO has better stability. The sensitivity analysis results of parameter θ indicate that as θ decreases, both objectives keep uptrend with fluctuations.

The BSS decision maker can use the nondominated optimal solutions of the biobjective BRP-BI in two ways. One method is to choose an optimal rebalanced scheme from the Pareto optimal set in the actual BSS operation. To this end, specific criteria of the decision maker must be introduced. Without these criteria, all solutions in the Pareto optimal set are not comparable. After setting the preferences for objectives, the decision maker can determine an optimal scheme from the Pareto optimal set. The other method is based on the above method. The nondominated optimal solutions allow the decision maker to balance the two conflicting objectives more flexibly. By weighing these objectives, the decision maker can quickly adjust the optimal scheme to adapt to complex and dynamic practical conditions. For example, if the total travel distance must be reduced, how much the rebalance utility can be sacrificed? Or, if the rebalance utility must be greater than a given value, what is the minimum travel distance required? From the nondominated optimal solutions, the decision maker can freely determine which optimal scheme suits the current situation. Compared with the monoobjective solution method, this is an obvious advantage of our MS-MOPSO.

Future research can extend our model and algorithm to solve the BRP-BI with multiple vehicles. In addition, dynamic or stochastic BRP with balance intervals is also a good research direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by a grant from Shanghai Planning Office of Philosophy and Social Science (2018BGL018).

References

- [1] Y. Chen, A. Bouferguene, H. X. Li, H. Liu, Y. Shen, and M. Al-Hussein, "Spatial gaps in urban public transport supply and demand from the perspective of sustainability," *Journal of Cleaner Production*, vol. 195, pp. 1237–1248, 2018.
- [2] R. Alvarez-Valdes, J. M. Belenguer, E. Benavent et al., "Optimizing the level of service quality of a bike-sharing system," *Omega*, vol. 62, pp. 163–175, 2016.
- [3] T. Bulhões, A. Subramanian, G. Erdoğan, and G. Laporte, "The static bike relocation problem with multiple vehicles and visits," *European Journal of Operational Research*, vol. 264, no. 2, pp. 508–523, 2018.
- [4] C. Fricker and N. Gast, "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity," *EURO Journal on Transportation and Logistics*, vol. 5, no. 3, pp. 261–291, 2016.
- [5] G. Erdoğan, G. Laporte, and R. Wolfler-Calvo, "The static bicycle relocation problem with demand intervals," *European Journal of Operational Research*, vol. 238, no. 2, pp. 451–457, 2014.
- [6] A. Pal and Y. Zhang, "Free-floating bike sharing: solving real-life large-scale static rebalancing problems," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 92–116, 2017.
- [7] 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, 2014.
- [8] S. Zhang, G. Xiang, and Z. Huang, "Bike-sharing static rebalancing by considering the collection of bicycles in need of repair," *Journal of Advanced Transportation*, vol. 2018, pp. 1–18, 2018.
- [9] F. Cruz, A. Subramanian, B. P. Bruck, and M. Iori, "A heuristic algorithm for a single vehicle static bike sharing rebalancing problem," *Computers & Operations Research*, vol. 79, pp. 19–33, 2017.
- [10] Y. Duan, J. Wu, and H. Zheng, "A greedy approach for vehicle routing when rebalancing bike sharing systems," *IEEE*, vol. 2018, 2018.
- [11] L. Shi, Y. Zhang, W. Rui, and X. Yang, "Study on the bike-sharing inventory rebalancing and vehicle routing for bike-

- sharing system,” *Transportation Research Procedia*, vol. 39, pp. 624–633, 2019.
- [12] M. Casazza, A. Ceselli, and R. Wolfler-Calvo, “A route decomposition approach for the single commodity split pickup and split delivery vehicle routing problem,” *European Journal of Operational Research*, vol. 264, 2019.
- [13] P. Papazek, C. Kloimüller, B. Hu, and G. R. Raidl, “Balancing bicycle sharing systems: an analysis of path relinking and recombination within a GRASP hybrid,” in *Parallel Problem Solving from Nature-PSN XIII*, T. Bartz-eielstein, Ed., pp. 792–801, Springer, Berlin, Germany, 2014.
- [14] L. Gaspero, A. Rendl, and T. Uri, “A hybrid ACO + CP for balancing bicycle sharing systems,” *Proceedings of the Eighth International Workshop on Hybrid Metaheuristics*, vol. 64, pp. 198–212, 2013.
- [15] T. Raviv and O. Kolka, “Optimal inventory management of a bike-sharing station,” *IIE Transactions*, vol. 45, no. 10, pp. 1077–1093, 2013.
- [16] W. Y. Szeto, Y. Liu, and S. C. Ho, “Chemical reaction optimization for solving a static bike repositioning problem,” *Transportation Research Part D: Transport and Environment*, vol. 47, pp. 104–135, 2016.
- [17] H. M. Espegren, J. Kristianslund, H. Andersson, and F. Kjetil, “The static bicycle repositioning problem -literature survey and new formulation,” *International Conference on Computational Logistics*, vol. 47, 2016.
- [18] S. C. Ho and W. Y. Szeto, “A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem,” *Transportation Research Part B: Methodological*, vol. 95, pp. 340–363, 2017.
- [19] T. Raviv, M. Tzur, and I. A. Forma, “Static repositioning in a bike-sharing system: models and solution approaches,” *EURO Journal on Transportation and Logistics*, vol. 2, no. 3, pp. 187–229, 2013.
- [20] S. C. Ho and W. Y. Szeto, “Solving a static repositioning problem in bike-sharing systems using iterated tabu search,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 69, pp. 180–198, 2014.
- [21] Y. Li, W. Y. Szeto, J. Long, and C. S. Shui, “A multiple type bike repositioning problem,” *Transportation Research Part B: Methodological*, vol. 90, pp. 263–278, 2016.
- [22] Y. Wang and W. Y. Szeto, “Static green repositioning in bike sharing systems with broken bikes,” *Transportation Research Part D: Transport and Environment*, vol. 65, pp. 438–457, 2018.
- [23] M. Usama, Y. Shen, and O. Zahoor, “Towards an energy efficient solution for bike-sharing rebalancing problems: a battery electric vehicle scenario,” *Energies*, vol. 12, no. 13, 2019.
- [24] D. Zhang, C. Yu, J. Desai, H. Y. K. Lau, and S. Srivathsan, “A time-space network flow approach to dynamic repositioning in bicycle sharing systems,” *Transportation Research Part B: Methodological*, vol. 103, pp. 188–207, 2017.
- [25] L. Caggiani, R. Camporeale, M. Ottomanelli, and W. Y. Szeto, “A modeling framework for the dynamic management of free-floating bike-sharing systems,” *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 159–182, 2018.
- [26] S. Yan, C.-C. Lu, and M.-H. Wang, “Stochastic fleet deployment models for public bicycle rental systems,” *International Journal of Sustainable Transportation*, vol. 12, no. 1, pp. 39–52, 2017.
- [27] M. Dell’Amico, M. Iori, S. Novellani, and A. Subramanian, “The bike sharing rebalancing problem with stochastic demands,” *Transportation Research Part B*, vol. 118, pp. 362–380, 2018.
- [28] F. Maggioni, M. Cagnolari, L. Bertazzi, and S. W. Wallace, “Stochastic optimization models for a bike-sharing problem with transshipment,” *European Journal of Operational Research*, vol. 276, no. 1, pp. 272–283, 2019.
- [29] M. Yuan, Q. Zhang, B. Wang, Y. Liang, and H. Zhang, “A mixed integer linear programming model for optimal planning of bicycle sharing systems: a case study in Beijing,” *Sustainable Cities and Society*, vol. 47, 2019.
- [30] B. Legros, “Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station,” *European Journal of Operational Research*, vol. 272, no. 2, pp. 740–753, 2019.
- [31] P.-S. You, “A two-phase heuristic approach to the bike repositioning problem,” *Applied Mathematical Modelling*, vol. 73, pp. 651–667, 2019.
- [32] J. X. Cao, C. C. Xue, M. Y. Jian, and X. R. Yao, “Research on the station location problem for public bicycle systems under dynamic demand,” *Computers & Industrial Engineering*, vol. 127, pp. 971–980, 2019.
- [33] J. Warrington and D. Ruchti, “Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems,” *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 110–134, 2019.
- [34] R. Regue and W. Recker, “Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 72, pp. 192–209, 2014.
- [35] A. A. Kadri, I. Kacem, and K. Labadi, “A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems,” *Computers & Industrial Engineering*, vol. 95, pp. 41–52, 2016.
- [36] B. Mohammadi and H. Shirouyehzad, “Developing a multi-commodity multi-period mathematical model based on the travelling salesman problem for solving bike sharing rebalancing problem,” *International Journal of Modelling in Operations Management*, vol. 7, no. 1, pp. 59–74, 2018.
- [37] W. Y. Szeto and C. S. Shui, “Exact loading and unloading strategies for the static multi-vehicle bike repositioning problem,” *Transportation Research Part B: Methodological*, vol. 109, pp. 176–211, 2018.
- [38] K. Liu, C. Zou, K. Li, and T. Wik, “Charging pattern optimization for lithium-ion batteries with an electrothermal-aging model,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5463–5474, 2018.
- [39] K. Liu, X. Hu, Z. Yang, Y. Xie, and S. Feng, “Lithium-ion battery charging management considering economic costs of electrical energy loss and battery degradation,” *Energy Conversion and Management*, vol. 195, pp. 167–179, 2019.
- [40] Q. Zhang and H. Li, “MOEA/D: a multi-objective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [41] N. Dabiri, M. J. Tarokh, and M. Alinaghian, “New mathematical model for the bi-objective inventory routing problem with a step cost function: a multi-objective particle swarm optimization solution approach,” *Applied Mathematical Modelling*, vol. 49, pp. 302–318, 2017.
- [42] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

- [43] S. Zhang, G. Xiang, and Z. Huang, "Bike-sharing static rebalancing by considering the collection of bicycles in need of repair," *Journal of Advanced Transportation*, vol. 2018, 2018.
- [44] O. Bräysy, G. Hasle, and W. Dullaert, "A multi-start local search algorithm for the vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 159, no. 3, pp. 586–605, 2004.
- [45] R. Baños, J. Ortega, C. Gil, A. L. Márquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 286–296, 2013.
- [46] B. Wu, W. Hu, J. Hu, and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on evolutionary state estimation," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2019.