Hindawi
Scientific Programming
Volume 2020, Article ID 8868892, 12 pages
https://doi.org/10.1155/2020/8868892

Research Article

Addressing the Bike Repositioning Problem in Bike Sharing System: A Two-Stage Stochastic Programming Model

Qiong Tang,[1,2,3] **Zhuo Fu**,[1,3] **Dezhi Zhang,**[1,3] **Hao Guo**,[4] **and Minyi Li**[5]

[1]*School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China*
[2]*College of Economics and Management, Hengyang Normal University, Hengyang 421002, China*
[3]*Smart Transport Key Laboratory of Hunan Province, Changsha 410075, China*
[4]*School of Management, Jinan University, Guangzhou 510632, China*
[5]*School of Science, RMIT University, Melbourne, VIC 3000, Australia*

Correspondence should be addressed to Zhuo Fu; zhfu@csu.edu.cn

Received 10 April 2020; Revised 26 April 2020; Accepted 5 May 2020; Published 20 May 2020

Academic Editor: Lu Zhen

In this paper, a bike repositioning problem with stochastic demand is studied. The problem is formulated as a two-stage stochastic programming model to optimize the routing and loading/unloading decisions of the repositioning truck at each station and depot under stochastic demands. The goal of the model is to minimize the expected total sum of the transportation costs, the expected penalty costs at all stations, and the holding cost of the depot. A simulated annealing algorithm is developed to solve the model. Numerical experiments are conducted on a set of instances from 20 to 90 stations to demonstrate the effectiveness of the solution algorithm and the accuracy of the proposed two-stage stochastic model.

1. Introduction

Bike sharing system (BSS) as a means of sustainable and carbon-free transportation can effectively solve the "the first and last mile" problem of urban public transportation. BSS has gained increasing popularity in recent years due to the fact that it not only reduces urban pollution emissions and traffic congestion but also is considered as an effective way to improve the health of users [1]. A BSS consists of a depot and several stations, which scatter in different streets of the city. The success of a BSS depends on the availability of bikes. Due to the rental cycling of the user, there are often unbalanced situations at some stations of the system, i.e., there are either surplus or not enough bikes at these stations. In other words, either when there are surplus bikes in a station, they will be wasted as no users will need them, or in some stations where there are not enough bikes, a user's demand cannot be met. To ensure the availability of bikes at all stations in the system, the BSS needs to employ redistribution trucks to transfer bikes from stations to stations, to balance the number of bikes according to the demand in each station. This problem is commonly defined as the bike repositioning problem (BRP) in the BSS. The main objective of the BRP is to balance the supply and demand of bikes across stations as a mean to improve user satisfaction, while in the meantime, to reduce the transportation cost of redistribution trucks as much as possible. According to the classification of Berbeglia et al. [2], the bike repositioning problem is essentially a many-to-many pickup and delivery problem, and it is first proven to be an NP-hard problem by Benchimol et al. [3].

Typically, existing work assumes the redistribution demand at each station in the BSS is deterministic input parameters [4–21]. That is to say, these models heavily rely on that the loading and unloading quantities of redistribution trucks at each station be given in advance. However, this strong assumption generally could not hold in real-world conditions where the demand in urban public transportation commonly contains uncertainty and randomness [22]. As pointed out by Fricker and Gast [23], the effect of the uncertainty in user demand should not be neglected when studying the performance of a BSS. The authors have also demonstrated that incorporating knowledge of future demands can lead to more accurate and robust decisions in repositioning.

The other challenge that might not have received enough attention in BRP is the consideration of the holding cost of the depot. In the BRP, bikes transferred from a bike surplus station to another bike deficient station is required to achieve the balance of supply and demand at all bike stations in the BSS to satisfy the user's demand. This perfect balance of supply and demand after repositioning operation must first ensure the internal balance in the BSS, i.e., the number of bikes at bike surplus stations should be equal to the number of bikes at bike deficient stations. In the actual operation process, some bikes might be lost or damaged after they are put into use. This happens quite often and will likely lead to the inherent imbalance between supply and demand in the BSS. For the bikes which cannot be balanced among the stations in the BSS, they are either picked up from or delivered back to the depot at the cost of increasing holding cost of the depot. Some BRP models have simply assumed the capacity of the depot is sufficient. As such, some bikes can be picked up from the depot to meet the demand of bike deficient stations, and/or some bikes at bike surplus stations can be brought back to the depot, in order to achieve a perfect balance between supply and demand at all stations in the BSS [4–7]. Unfortunately, these existing works have not yet considered the holding cost of the depot in the BRP model.

Stochastic programming is a technology to design mathematical programming models with stochastic demand, which helps decision makers to make more accurate decisions when the effect of the randomness of demand cannot be ignored. Although the optimization results obtained from stochastic programming cannot satisfy all future scenarios, they help to achieve better-than-average performance in dealing with stochastic demand.

To address the abovementioned concerns, in this paper, we take into account the stochastic nature of the BRP by capturing and modeling the uncertainty of redistribution demand within the system using stochastic programming. In particular, we propose a two-stage stochastic programming model for the bike repositioning problem with stochastic demand (BRPSD). In the first stage, we determine the routing decision, where "here-and-now" decision must be made before the realization of redistribution demand is known. Then, moving to the second stage, we further determine the loading/unloading decisions at each station and depot, where "wait-and-see" decisions are made taking future uncertainty into account. We incorporate the holding cost into the model such that the overall objective of the BRPSD is to determine the best routes of the repositioning truck and the optimal loading/unloading quantities at each station and depot, such that the expected total sum of the transportation costs, the penalty costs at all stations, and holding cost of the depot are minimized.

In this paper, stochastic redistribution demands are modeled by a set of discrete scenarios, and a predefined probability of occurrence is given for each scenario. The main contributions of this paper are summarized as follows:

(1) A general scenario-based two-stage stochastic programming model for the bike repositioning problem with stochastic demands (BRPSD) is introduced. The proposed model takes the inventory holding cost of the depot into account in contexts where the loading and unloading activities are allowed in the depot.

(2) An efficient metaheuristic, namely, simulated annealing algorithm is proposed to solve the defined BRPSD.

(3) The necessity of considering redistribution demand as a stochastic demand in the BRPSD is further discussed with numerical experiments.

The remainder of this paper is organized as follows. We review the relevant literature in Section 2 and describe the mathematical formulation of the problem in Section 3. In section 4, we present the improved simulated annealing algorithm to solve the model. Then, we present experimental results in Section 5 and conclude this paper with discussions on potential future research in Section 6.

2. Literature Review

To the best of our knowledge, Benchimol et al. first formalise and address the BRP in [3]. Since then, various models and solution algorithms for the BRP have been discussed over the last decades.

We briefly review studies that are related to this paper, including heuristic and metaheuristic solution methods, for solving the deterministic BRP and models for the BRP with stochastic demands.

2.1. Heuristic Solution Algorithm for the Deterministic BRP.

According to Pal and Zhang [8], the BRP can be categorized into complete and partial repositioning problems according to the rigor of a repositioning that needs to be performed. In the former problem, the constraint for achieving a perfect balance among stations is considered as a hard constraint, and in the latter problem, it is a soft constraint.

In the complete repositioning problem, the optimal inventory of each station must be met after the repositioning operation. An iterated local search algorithm was first applied by Cruz et al. [9] to solve the single-vehicle BRP and then used by Bulhões et al. [10] to solve the multivehicle version of the BRP. Later, Pal and Zhang [8] modeled a static BRP in a free-floating bike sharing system and presented a hybrid nested large neighborhood search with a variable neighborhood descent algorithm to tackle the BRP. Dell'Amico et al. [11] developed a destroy and repair metaheuristic algorithm, which makes use of effective constructive heuristic and several local search procedures together for solving the BRP. Overall, the objective of the complete repositioning problem is mainly to pursue the minimum transportation cost by determining routing decisions. It has been a challenge to apply such methods in real-world systems effectively as there are other equally important criteria that have been neglected in these models.

In the partial repositioning problem, the ideal inventory of each station is not necessarily achieved after the repositioning operation. Nonetheless, the goal of these problems is not only to pursue the minimum transportation cost by determining routing decisions but also to minimize user dissatisfaction by determining loading and unloading decisions at each station. In the literature, a 3-step math heuristic for the multivehicle BRP is proposed by Forma et al. [6]. In their approach, stations are firstly clustered by a saving heuristic. Then, the routes through the clusters are constructed. Finally, the routing sequence through the stations in each cluster and the sequence through the clusters are determined. The last two steps are formulated as a mixed integer linear model solving by CPLEX. You [12] proposes a two-phase heuristic for the multivehicle BRP under a minimum service requirement over planning. In his proposed model, each stage represents a decision process. In the first phase, loading and unloading decisions for all stations for each time slot are developed by a linear programming model. Then, in the second phase, routing decisions are developed by an iterative approach through two parameter sensitive mathematical models.

In addition to the phased heuristic algorithm, many heuristics and metaheuristic solution methods have been presented to solve the deterministic BRP, to name a few, tabu search [5,13], genetic algorithm [14], chemical reaction optimization algorithm [15,16], artificial bee colony algorithm [17], constraint programming [18,19], variable neighborhood search [20], iterated local search [21], large neighborhood search algorithm [7], etc. Nonetheless, the practical applications of these existing approaches remain difficult and often bring in unexpected results. This is because uncertainty about user demands commonly exists in real-world BRPs; however, it is not yet captured by such deterministic models.

2.2. The BRP with Stochastic Demands. To the best of our knowledge, limited literature considers stochastic demands in the BRP. Two-stage stochastic programming with recourse models have been successfully used in solving bike repositioning problems as they allow the modeler to represent routing plans and loading and unloading activities together via first- and second-stage decision variables. Dell'Amico et al. [24] appear to be the pioneer paper that proposed a scenario-based two-stage stochastic model for bike repositioning problem, considering redistribution demands of stations as random variables. The authors provide five exact procedures as well as a heuristic algorithm that combines correlation-based constructive procedures with VND local search approach to solve their BRPSD. The heuristic algorithm only considers transportation costs while evaluating the local search move and only accept a feasible solution where the feasibility of the local search move is checked by an efficient strategy.

However, the BRPSD differs from our problem in the following three aspects. First, the objective of this model is to minimize the expected total travel distance of the truck and the penalty costs at all stations, while our problem minimizes the expected total travel cost of the truck, the penalty costs at all stations, and the handling cost at the depot. Second, in our problem, we measure the local search move by using not only transportation costs but also the expected total costs. Third, we develop a method to quickly determine the feasible loading and unloading quantities at all stations and depot under any given route. Therefore, it is not necessary to check the feasibility of the solution in the local search.

There are also a few recent works that study a variation of the stochastic BRP where the stochastic demand is defined by origin-destination (OD) pair rather than the redistribution of stations. For instance, Maggioni et al. [25] propose two-stage stochastic programming models to determine the optimal number of bikes to assign to each station and the optimal number of transshipped bikes from one station to another station, respectively. They use AMPL and CPLEX to solve the model. Yan et al. [26] apply the time-space network model to determine the optimal locations of bike stations, bike fleet allocation, and bike routing. Their solution algorithm is based on a threshold-accepting-based heuristic.

From the abovementioned review of literature, we can see that the current scholars have proposed a rich heuristic algorithm to solve the deterministic BRP. Nonetheless, to the best of our knowledge, there is no work yet that effectively employs a metaheuristic algorithm to solve the BRP with stochastic demand. The study of this paper fills this theoretical gap.

3. Mathematical Formulation

The bike sharing system studied in this paper comprises a depot and a set of stations. Each station has a redistribution demand for each scenario, indicating the difference between the current inventory level and the optimal inventory level for each scenario. For a certain scenario, if the redistribution demand at a station is positive, the station is defined as a pickup station; if the redistribution demand at a station is negative, then the station is defined as a delivery station and should be supplied with bikes from pickup stations; if the redistribution demand at a station is equal to 0, then the station is called a balanced station. Balanced stations must also be visited.

We consider that only a single truck with a given capacity is available in the BSS. The truck collects bikes from pickup stations and transports them to delivery stations. Some bikes can be picked up from or delivered back to the depot, and each station is allowed to be visited only once. In addition, similar to Liu et al. [16], bikes are assumed to be parked anywhere in the BSS; therefore, the station capacity constraint is removed.

The decisions regarding the loading and unloading quantities at all stations and the depot in each scenario are made independently, according to the corresponding demand scenario. However, the routing decisions of the truck have to take into consideration all possible demand scenarios. This means that routing decisions are the same for all scenarios. Because the routing decisions to all scenarios are the same but the loading and unloading decisions vary in each scenario; there may exist a shortage or excess of bikes for each station in each scenario. In this context, the following two-stage bike repositioning problem with stochastic demands (BRPSD) naturally arises. In the first stage, the BRPSD aims to construct a route for the truck that starts from and ends at the depot where each station is visited only once. In the second stage, the BRPSD aims to determine the optimal loading and unloading quantities at all stations and the depot to ensure the feasibility of the solution after the given route in the first stage. The objective of the BRPSD is to minimize the sum of the travel costs for the truck (first-stage objective function) and the expected total penalty costs at all stations and holding costs at the depot (second-stage objective function).

3.1. Two-Stage Stochastic Programming Model. Our objective is to determine the sequence of all stations to visit to minimize the transportation cost, as well as the loading/unloading decisions in each scenario to minimize the total expected penalty costs and holding costs. We formulate the problem as a two-stage stochastic programming problem. Given the discrete probability distribution of the scenario occurrences, the formulation for the SPBSD addressed in this study is as follows:

$$\min \sum_{i\in N_0} \sum_{j\in N_0, i\neq j} C_t t_{ij} x_{ij} + \sum_{\xi\in\Xi} p^{\xi}\left(\sum_{i\in N}\left(C_d\left(s_i^{\xi+}+s_i^{\xi-}\right)+C_h\left(y_{0i}^{\xi}+y_{i0}^{\xi}\right)\right)\right), \tag{1}$$

subject to

$$\sum_{j\in N} x_{0j} = \sum_{j\in N} x_{j0} = 1, \tag{2}$$

$$\sum_{j\in N_0, j\neq i} x_{ij} = 1, \quad i\in N, \tag{3}$$

$$\sum_{j\in N_0, j\neq i} x_{ij} = \sum_{j\in N_0, j\neq i} x_{ji}, \quad i\in N, \tag{4}$$

$$q_j \geq q_i + 1 - M\left(1-x_{ij}\right), \quad \forall i, j\in N_0, i\neq j, \tag{5}$$

$$y_{ij}^{\xi} \leq Q x_{ij}, \quad \forall \xi\in\Xi, i\in N_0, j\in N_0, i\neq j, \tag{6}$$

$$\sum_{j\in N_o, j\neq i} y_{ij}^{\xi} - \sum_{j\in N_o, j\neq i} y_{ji}^{\xi} = d_i^{\xi} + s_i^{\xi-} - s_i^{\xi+}, \quad \forall \xi\in\Xi, i\in N, \tag{7}$$

$$q_i \geq 0, \quad \forall i\in N_0, \tag{8}$$

$$s_i^{\xi-}, s_i^{\xi+} \geq 0, \quad \forall i\in N, \xi\in\Xi, \tag{9}$$

$$y_{ij}^{\xi} \geq 0, \quad \forall i, j\in N_0, i\neq j, \xi\in\Xi, \tag{10}$$

$$x_{ij}\in\{0,1\}, \quad \forall i, j\in N_0, i\neq j. \tag{11}$$

The objective function (1) is defined as the weighted sum of the transportation costs and the expected penalty costs at stations and holding costs at the depot over all scenarios. Constraint (2) states that the truck departs from the depot only once and returns to the depot. Constraint (3) ensures that the truck can visit a station only once. Constraint (4) ensures that if the truck visits a station, it must leave that station. Constraint (5) is the subtour elimination constraint [27]. Constraint (6) ensures that the load on the vehicle cannot be greater than the truck capacity. Constraint (7) requires that the number of bikes loaded onto or unloaded from the truck at a given station equals the difference between the truck load before and after the station visits. Constraint (8) ensures the auxiliary variable is nonnegative. Constraints (9)-(10) ensure that the excess/slack quantities of bikes at station and the number of bikes on the truck are nonnegative integers. Constraint (11) defines x_{ij} to be a binary variable.

4. Solution Algorithm

When the number of scenarios equals 1, the studied BRPSD reduces to the classical deterministic bike repositioning problems, an NP-hard combinatorial optimization problem. We employ simulated annealing (SA) to solve the studied BRPSD. SA is a metaheuristic algorithm based on local search, which can avoid falling into local optimization by accepting poor solutions with less probability in the process of iteration. It has been successfully applied to a variety of deterministic combinatorial problems [28,29] and stochastic combinatorial problems [30,31].

Our proposed algorithm, called $\mathrm{SA}_{\mathrm{BRPSD}}$, consists of two phases: constructing the initial solution and improving the initial solution by neighborhood search mechanism. The structure of this section is organized as follows. In Section 4.1, we propose the heuristic method to generate an initial solution. During initialization, two main decisions must be made: the routing decision (in Section 4.1.1) and the loading and unloading decision (i.e., inventory decision) (in Section 4.1.2). Then, in Section 4.2, we describe in detail of the local search operators used in this $\mathrm{SA}_{\mathrm{BRPSD}}$, namely, Swap, Relocate, and 2-opt. After that the main algorithm flow based on the $\mathrm{SA}_{\mathrm{BRPSD}}$ framework is presented in Section 4.3.

4.1. Construct Initial Solution. Since the BRPSD is a two-stage decision-making process, the decision-making in this paper is divided into two stages: routing decisions are determined in the first stage, and the loading and unloading decisions are determined in the second stage. The process of constructing the initial solution is as follows, which is roughly divided into two main phases.

4.1.1. Construct the Initial Route for all Scenarios. Since the route in all scenarios is the same, the greedy heuristic algorithm is used to quickly construct the initial route by ignoring the stochastic redistribution demand. In this work, the nearest principle is used to construct the initial route by inserting stations one by one.

4.1.2. Determine Loading and Unloading Decision for Each Scenario. For a given scenarioξ and the corresponding redistribution demand d_i^{ξ} at station i in the scenario, the **Second-Stage-Opt** procedure uses a simple principle to quickly generate the initial feasible loading and unloading decision and then uses a reoptimize heuristic operator to improve the initial loading and unloading decision. The details of the **Second-Stage-Opt** procedure are given as follows.

Step 1: determine the initial loading and unloading quantities of all stations and the depot. The loading and unloading quantities at each station are defined as the corresponding redistribution demand. Because the demand of the depot is 0, the loading and unloading quantities are also defined as 0. As a result, the initial solution of the BRPSD is generated, including two-stage decisions, namely, the routing decision and the loading and unloading decision. It is obvious that the initial solution is often not feasible.

Step 2: adjust the loading and unloading quantities of some stations to achieve a feasible initial solution. If the number of bikes on the truck when it travels directly from station i to station j in scenario$\xi y_{ij}^{\xi} > Q$ or $y_{ij}^{\xi} < 0$, then the initial solution is not feasible. In this case, the loading and unloading quantity of station i should be adjusted. Set$\gamma = y_{ij}^{\xi} - Q$, if $y_{ij}^{\xi} > Q$ or $\gamma = y_{ij}^{\xi}$, if $y_{ij}^{\xi} < 0$, then the loading and unloading quantity of station i is adjusted to subtract $\gamma$from the original quantity.

Step 3 (reoptimize): the basic idea of the reoptimize operator is to optimize the solution by a heuristic method that adjusts the loading and unloading quantities between stations and the depot [5,7]. In this paper, some bikes can be collected from the depot to meet the delivery demands of some stations, and also some bikes collected from some stations can be delivered to the depot at the end of the repositioning. In this procedure, there are two cases. In the first case, the truck picks up extra bikes when it starts from the depot and drops them off at some stations. In the second case, the truck picks up more bikes from some stations and drops them off when it ends at the depot.

4.2. Neighborhood Structure. In this paper, the local search operator aims at routing decisions. Three kinds of classical operators Swap move, Relocate move, and 2-opt move are used. These operators are also often used in other BRP problems. Each new neighborhood solution is generated by these three moves with equal probability. Note that, after a local search operator is executed, the new solution needs to be improved by *Second-Stage-Opt* procedure to obtain the second-stage decision.

Swap-N^1: randomly select two stations from the route and then exchange them

Relocate-N^2: randomly select one station, delete it from the current route, and then insert it into another position

2-Opt-N^3: randomly select two stations, exchange the positions of the two stations, and change the orientation of the route between the two stations

4.3. The Proposed SA$_{BRPSD}$. Algorithm 1 provides the pseudocodes of the SA$_{\text{BRPSD}}$. Parameter T_0 denotes the initial temperature; T_E is the final temperature; α is the coefficient used to control the cooling; I_{iter} indicates the number of generated new solutions under a certain temperature; and Max$_{\text{iter}}$ is the max number of the accepted new solutions under a certain temperature. In this section, we use S_0 and S^{best} to denote the initial solution and the best solution obtained so far, respectively. The best objective value $F(S^{\text{best}})$ is set to be the objective function value of the best solution S^{best}.

As shown in Algorithm 1, in the inner loop of the SA$_{\text{BRPSD}}$ algorithm, a random new neighborhood solution S^{new} is initially generated by *Local search* procedure to improve the first-stage decision and then followed by *Second-Stage-Opt* procedure to generate the second-stage decision. The stopping criteria for the inner loop are either it has generated I_{iter} new solutions or accepted Max$_{\text{iter}}$ new solution under the current temperature T. When the current temperature T decreases to T_E, the SA$_{\text{BRPSD}}$ algorithm terminates.

5. Numerical Examples

In this section, numerical examples are conducted to examine the validity and efficiency of the proposed model and solving algorithm. In addition, the necessity of considering stochastic demands is examined by analyzing the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS).

In this study, we consider the instances as in [24] to conduct the numerical experiments (the instances are available at http://www.or.unimore.it/site/home/online-resources). The size of the instances ranges from 20 to 90 stations. There are a total of 20 instances and 30 scenarios for each instance. In any instance, the first station is chosen to be the depot. To better analyze the impact of relevant parameters on the solution of the stochastic model, for a given set N_0, station i and station j belong the set $N_{0,}$ and we redefine the parameter transportation time t_{ij} as $t_{ij} = t_{ij}/\min\{t_{ij}\}$.

The SA$_{\text{BRPSD}}$ was coded in Matlab, and all computational experiments were carried out on a computer with an Intel Core i5-4590 CPU @ 3.30 GHz and 4 GB RAM. Each instance was run 10 times, the best and average of the solutions, and the average computing time of 10 runs were used to evaluate the performance of the algorithm solution.

The SA$_{\text{BRPSD}}$ proposed in this study relies on five parameters, namely, the initial temperature T_0; the final temperature T_E; the coefficient used to control the cooling α; the number of generated solutions under a certain temperature I_{iter}; and the number of accepted solutions under a certain temperature Max$_{\text{iter}}$. After some preliminary experiments, the results from the algorithm were obtained by

```
Obtain the initial solution $S_{0.}$
Set the parameter for the simulated annealing, $T_0$, $T_E$, $\alpha$, $I_{iter}$, $Max_{iter}$
Set $S^{best} \longleftarrow S_0$; $S^{current} \longleftarrow S_0$; $T \longleftarrow T_{0.}$
While $T \geq T_E$do
  $m \longleftarrow 0$; $n \longleftarrow 0$;
  While $m \leq I_{iter}$ & $n \leq Max_{iter}$ do
  $S \longleftarrow$ Local search ($S^{current}$, $N^k$);
  $S^{new} \longleftarrow$ Second-Stage-Opt ($S$);
    $\Delta \longleftarrow F(S^{new}) - F(S^{current})$;
  If $\Delta \leq 0$ then
    $S^{current} \longleftarrow S^{new}$, $n \longleftarrow n + 1$;
    If $F(S^{current}) \leq F(S^{best})$ then
      $S^{best} \longleftarrow S^{current}$;
    end
  else
    If $e^{-\Delta/T} \geq$ random then
      $S^{current} \longleftarrow S^{new}$, $n \longleftarrow n + 1$;
    end
  end
  $m \longleftarrow m + 1$;
  end
  $T \longleftarrow \alpha \square T$;
end
Return: $S^{best}$
```

Algorithm 1: SA_{BRPSD}.

setting $T_0 = 20$, $T_E = 0.1$, $\alpha = 0.97$, $I_{iter} = 3 * |N_0|$, and $Max_{iter} = |N_0|$.

5.1. The Performance Comparison between Lingo and SA Algorithm. In terms of the quality of the solution and the calculation efficiency, Tables 1–4 show the comparison results obtained by Lingo and SA_{BRPSD} under different truck capacity Q, the holding cost C_h, and the number of stations. The CPU (in seconds) indicates the average computing time of the SA_{BRPSD} or the computing time for Lingo, respectively. Gap_{Best} (%) and Gap_{Avg} (%) indicate the performance of the SABRP relative to that of Lingo based on the best and average objective values, respectively. When $|N_0| \geq 50$, Lingo is not able to obtain any feasible solutions for the problems, and thus, the results are not shown in the tables. It also generally fails to obtain any global optimal solutions within 3600 seconds even for small-size problems (with four exceptions for the results obtained under instance *Washington (20)2*). From Tables 1–4, we can see that SA_{BRPSD} is significantly faster than Lingo in all instances, i.e., it has better time efficiency to solve the BRPSD compared to Lingo. For the various sizes of problems from small to large, the average computing time of the SA_{BRPSD} varies from 66.14 s to 70.22 s.

As shown in Tables 1–4, all of the values of Gap_{Best} and Gap_{Avg} are negative. This implies that the SA_{BRPSD} performs better than Lingo. The absolute values of the average Gap_{Best} and Gap_{Avg} values obtained by the SA_{BRPSD} for the tests with $Q = 20$ are larger than those with $Q = 10$. As the capacity of the repositioning truck increases, the objective function value decreases. This may be explained by the fact that if the truck capacity increases, the truck can load/unload more bikes at one station, resulting in an expansion of the solution space. Overall, we can draw the conclusion that the proposed SA_{BRPSD} can obtain better feasible optimal solutions much faster than Lingo.

5.2. The Necessity of Considering Demand as a Stochastic Parameter. To evaluate the performance of the stochastic model we proposed in this paper, we use the two concepts of the evaluation approach: the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) [32–34]. EVPI measures the maximum amount a decision maker would be ready to pay in return for perfect information about the future, and VSS measures the amount of cost saving when the decision maker uses expected values of stochastic parameters instead of stochastic parameters in the model (the amount of cost due to ignoring the uncertainty). EVPI is defined as the difference between the wait-and-see solution (WS) and the here-and-now solution (HN). In the EVPI, the WS solution is the expected value of the foresight. We obtain the value by averaging the results of each stochastic scenario. The HN solution is the objective value of this two-stage stochastic model. VSS is defined as the difference between the expected results of using the expected value problem solution (EEV) and the here-and-now solution. In the VSS, the value of EEV is obtained by inputting the optimal solution of the optimization problem associated with the mean value of stochastic demands into the two-stage stochastic model. The gaps Gap_{VSS} and Gap_{EVPI} for VSS and EVPI are calculated by VSS/HN and EVPI/WS, respectively.

Table 1: Comparison of the results between Lingo and $\mathrm{SA}_{\mathrm{BRPSD}}$ ($Q = 20$ and $C_h = 1$).

Instance	Lingo		$\mathrm{SA}_{\mathrm{BRPSD}}$			$\mathrm{Gap}_{\mathrm{Best}}$ (%)	$\mathrm{Gap}_{\mathrm{Avg}}$ (%)
	Optimal value	CPU	Best value	Average value	CPU		
Washington (20)1	639.15	3600	639.16	639.17	10.15	0.00	0.00
Washington (20)2	576.79	159	576.79	576.79	24.58	0.00	0.00
Chicago (20)1	166.24	3600	153.40	153.40	20.85	−7.72	−7.72
Chicago (20)2	115.19	3600	113.46	113.46	14.04	−1.50	−1.50
Washington (30)1	450.19	3600	387.78	388.03	30.77	−13.86	−13.81
Washington (30)2	520.39	3600	403.12	405.09	25.22	−22.53	−22.16
Chicago (30)1	518.02	3600	265.71	266.96	54.21	−48.71	−48.47
Chicago (30)2	204.14	3600	202.21	204.14	29.48	−0.94	0.00
Washington (40)1	797.87	3600	634.07	638.52	44.71	−20.53	−19.97
Washington (40)2	1036.51	3600	826.68	831.62	29.94	−20.24	−19.77
Chicago (40)1	1287.45	3600	424.34	424.34	52.13	−67.04	−67.04
Chicago (40)2	478.01	3600	213.10	215.92	48.37	−55.42	−54.83
Washington (50)1			546.18	554.30	105.62		
Washington (50)2			603.40	611.72	77.76		
Chicago (50)			326.32	328.59	70.84		
Washington (66)			682.51	685.62	101.87		
Chicago (66)			6243.75	6494.41	122.49		
Washington (80)1			719.65	732.09	121.20		
Washington (80)2			678.58	684.66	159.47		
Washington (90)			925.21	938.56	196.93		
Avg		**3335**			**66.14**	**−24.15**	**−23.85**

Table 2: Comparison of the results between Lingo and $\mathrm{SA}_{\mathrm{BRPSD}}$ ($Q = 10$ and $C_h = 1$).

Instance	Lingo		$\mathrm{SA}_{\mathrm{BRPSD}}$			$\mathrm{Gap}_{\mathrm{Best}}$ (%)	$\mathrm{Gap}_{\mathrm{Avg}}$ (%)
	Optimal value	CPU	Best value	Average value	CPU		
Washington (20)1	668.94	3600	662.46	662.50	12.63	−0.97	−0.96
Washington (20)2	576.79	663	576.79	576.79	24.63	0.00	0.00
Chicago (20)1	164.14	3600	164.14	164.14	19.19	0.00	0.00
Chicago (20)2	136.83	3600	135.19	135.22	16.13	−1.20	−1.17
Washington (30)1	447.37	3600	393.75	393.75	34.19	−11.99	−11.99
Washington (30)2	501.14	3600	414.52	417.24	28.02	−17.28	−16.74
Chicago (30)1	300.48	3600	299.42	300.05	34.42	−0.35	−0.14
Chicago (30)2	299.76	3600	242.00	242.69	33.23	−19.27	−19.04
Washington (40)1	736.88	3600	688.63	692.12	60.31	−6.55	−6.07
Washington (40)2	911.88	3600	855.64	863.20	38.21	−6.17	−5.34
Chicago (40)1	576.94	3600	449.41	449.41	47.47	−22.10	−22.10
Chicago (40)2	538.80	3600	243.16	245.98	50.99	−54.87	−54.35
Washington (50)1			581.86	588.84	85.98		
Washington (50)2			647.38	650.63	75.82		
Chicago (50)			353.99	354.51	69.17		
Washington (66)			676.54	683.93	90.98		
Chicago (66)			6192.78	6468.38	120.21		
Washington (80)1			733.58	744.35	114.69		
Washington (80)2			736.77	745.27	183.02		
Washington (90)			984.90	996.38	228.76		
Avg		**3355**			**68.40**	**−11.73**	**−11.49**

The test instance *Chicago 20(1)* used for explaining the total costs of WS, HN, EEV, EVPI, and VSS is a small-size instance consisting of 20 stations. We set the truck capacity $Q = 10$, $C_h = 1$, $C_d = 1$, and $C_t = 1$.

In this test instance, the total costs can be solved by the $\mathrm{SA}_{\mathrm{BRPSD}}$ as follows: WS = 145.31, HN = 153.40, and EEV = 161.31. Then, calculate VSS = EEV − HN = 7.90, EVPI = HN − WS = 8.09, $\mathrm{Gap}_{\mathrm{VSS}} = 5.15\%$, and $\mathrm{Gap}_{\mathrm{EVPI}} = 5.57\%$. Both values of EVPI and VSS lead to the conclusion that managers must pay 5.57% more of the total cost of the stochastic solutions to obtain the perfect redistribution demand information about the future. Ignoring the uncertainty will let managers in a situation where they have a risk to pay at least 5.15% higher if they only focus on a scenario only (deterministic solutions).

The results of the sensitivity analysis on $\mathrm{Gap}_{\mathrm{VSS}}$ and $\mathrm{Gap}_{\mathrm{EVPI}}$ for the truck capacity Q, the holding cost C_h, the penalty cost C_d, and the travel cost C_t are explained in the

Table 3: Comparison of the results between Lingo and SA_{BRPSD} ($Q = 20$ and $C_h = 0$).

Instance	Lingo		SA_{BRPSD}			Gap_{Best} (%)	Gap_{Avg} (%)
	Optimal value	CPU	Best value	Average value	CPU		
Washington(20)1	645.10	3600	627.85	627.90	13.76	−2.67	−2.67
Washington(20)2	556.26	40	556.26	556.26	33.82	0.00	0.00
Chicago(20)1	134.73	3600	134.73	134.73	20.53	0.00	0.00
Chicago(20)2	100.80	3600	99.60	99.76	17.95	−1.19	−1.04
Washington(30)1	414.18	3600	369.72	369.72	32.90	−10.73	−10.73
Washington(30)2	432.25	3600	392.72	393.61	27.89	−9.15	−8.94
Chicago(30)1	489.71	3600	249.47	249.80	35.62	−49.06	−48.99
Chicago(30)2	321.25	3600	189.40	191.21	31.73	−41.04	−40.48
Washington(40)1	721.11	3600	617.41	622.76	50.49	−14.38	−13.64
Washington(40)2	1021.87	3600	814.77	819.79	42.10	−20.27	−19.78
Chicago(40)1	412.36	3600	408.94	408.94	57.98	−0.83	−0.83
Chicago(40)2	397.27	3600	201.52	203.18	50.88	−49.27	−48.86
Washington(50)1			526.11	534.88	109.26		
Washington(50)2			592.72	595.78	79.45		
Chicago(50)			321.13	324.60	69.87		
Washington(66)			656.99	663.09	105.25		
Chicago(66)			6243.75	6494.41	122.49		
Washington(80)1			702.93	716.69	126.66		
Washington(80)2			661.88	673.31	171.68		
Washington(90)			914.40	924.30	204.03		
Avg		**3303.33**			**70.22**	**−16.55**	**−16.33**

Table 4: Comparison of the results between Lingo and SA_{BRPSD} ($Q = 10$ and $C_h = 0$).

Instance	Lingo		SA_{BRPSD}			Gap_{Best} (%)	Gap_{Avg} (%)
	Optimal value	CPU	Best value	Average value	CPU		
Washington(20)1	665.08	3600	654.91	655.01	10.76	−1.53	−1.52
Washington(20)2	566.26	30	566.26	566.26	25.00	0.00	0.00
Chicago(20)1	155.17	3600	155.17	155.17	19.20	0.00	0.00
Chicago(20)2	127.97	3600	127.97	127.97	17.38	0.00	0.00
Washington(30)1	449.66	3600	382.56	382.56	33.10	−14.92	−14.92
Washington(30)2	442.69	3600	410.45	411.18	28.40	−7.28	−7.12
Chicago(30)1	292.28	3600	289.57	290.67	33.25	−0.93	−0.55
Chicago(30)2	236.60	3600	234.10	234.10	32.92	−1.06	−1.06
Washington(40)1	818.25	3600	681.61	685.40	61.54	−16.70	−16.24
Washington(40)2	859.04	3600	853.74	857.84	47.10	−0.62	−0.14
Chicago(40)1	500.44	3600	442.74	442.74	47.85	−11.53	−11.53
Chicago(40)2	349.26	3600	239.51	240.61	53.41	−31.42	−31.11
Washington(50)1			571.78	579.44	106.77		
Washington(50)2			638.40	640.92	75.32		
Chicago(50)			352.67	354.72	72.96		
Washington(66)			664.31	671.87	89.90		
Chicago(66)			6175.14	6472.54	117.22		
Washington(80)1			728.64	739.63	114.94		
Washington(80)2			722.88	743.14	183.62		
Washington(90)			984.90	996.38	228.76		
Avg		**3303**			**69.97**	**−7.17**	**−7.02**

Table 5: Results of different vehicle capacities (Q).

Q	WS	HN	EEV	VSS	Gap_{VSS} (%)	EVPI	Gap_{EVPI} (%)
10	153.86	164.14	168.57	4.43	2.70	10.28	6.68
15	147.66	156.87	163.24	6.37	4.06	9.22	6.24
20	145.31	153.40	161.31	7.90	5.15	8.09	5.57
25	144.69	151.60	160.97	9.37	6.18	6.91	4.78
30	144.35	150.87	160.41	9.54	6.32	6.53	4.52
Avg				7.52	4.88	8.21	5.56

Table 6: Results of different holding costs (C_h).

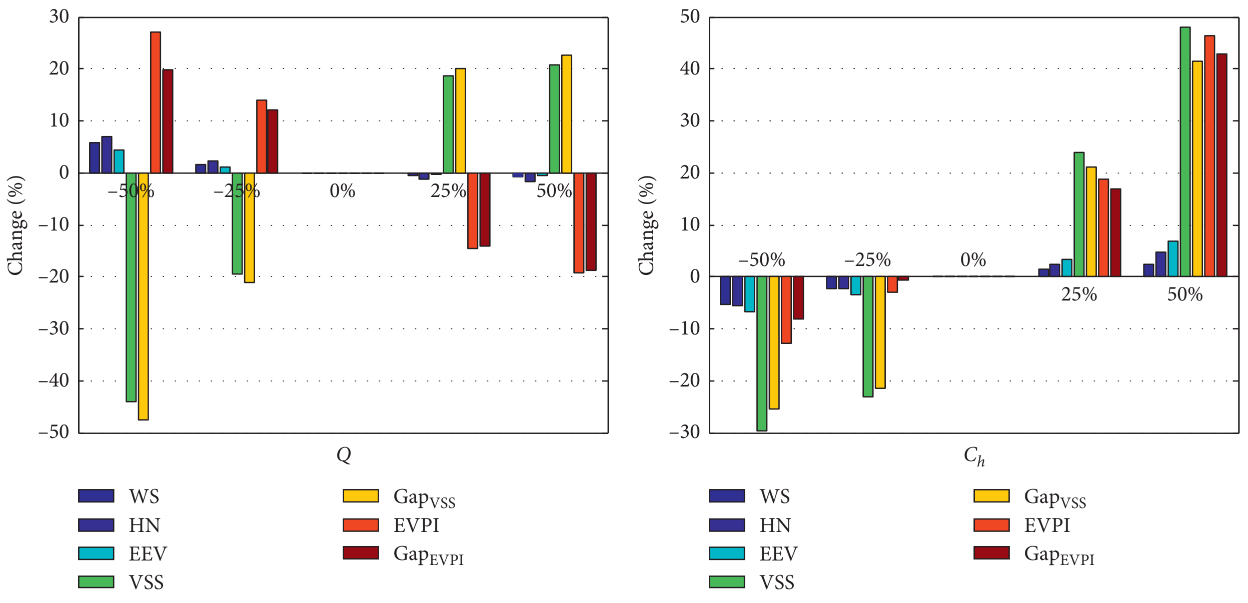
C_h	WS	HN	EEV	VSS	Gap_{VSS} (%)	EVPI	Gap_{EVPI} (%)
0.5	137.68	144.73	150.29	5.56	3.84	7.05	5.12
0.75	141.89	149.73	155.80	6.07	4.05	7.84	5.53
1	145.31	153.40	161.31	7.90	5.15	8.09	5.57
1.25	147.43	157.02	166.82	9.79	6.24	9.60	6.51
1.5	148.80	160.64	172.32	11.69	7.28	11.84	7.96
Avg				8.20	5.31	8.91	6.16

Table 7: Results of different penalty costs (C_d).

C_d	WS	HN	EEV	VSS	Gap_{VSS} (%)	EVPI	Gap_{EVPI} (%)
0.5	120.92	125.00	128.24	3.24	2.59	4.08	3.38
0.75	133.14	139.57	144.77	5.21	3.73	6.43	4.83
1	145.31	153.40	161.31	7.90	5.15	8.09	5.57
1.25	157.43	167.19	177.84	10.65	6.37	9.76	6.20
1.5	169.50	180.97	194.37	13.40	7.41	11.47	6.77
Avg				8.08	5.05	7.97	5.35

Table 8: Results of different travel costs (C_t).

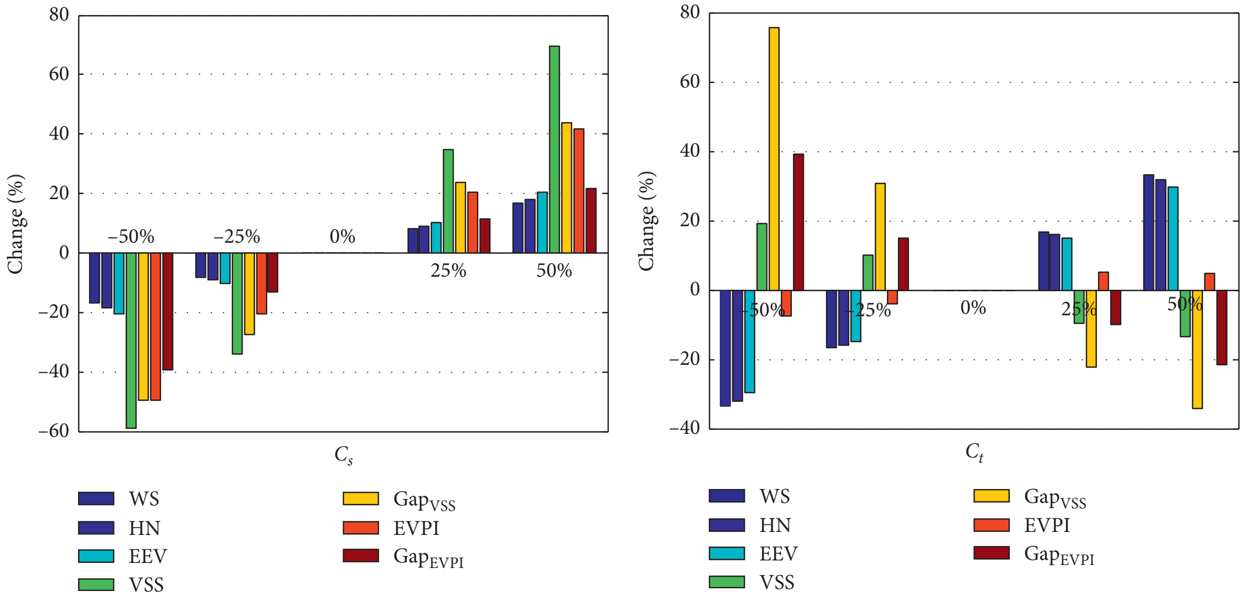
C_t	WS	HN	EEV	VSS	Gap_{VSS} (%)	EVPI	Gap_{EVPI} (%)
0.5	96.81	104.30	113.72	9.42	9.04	7.49	7.74
0.75	121.09	128.84	137.51	8.68	6.74	7.75	6.40
1	145.31	153.40	161.31	7.90	5.15	8.09	5.57
1.25	169.49	177.97	185.10	7.13	4.01	8.49	5.01
1.5	193.61	202.07	208.90	6.83	3.38	8.47	4.37
Avg				7.99	5.66	8.06	5.82



(a)

(b)

Figure 1: Continued.



(c)

(d)

Figure 1: Sensitivity analysis on different values of parameters: (a) Q; (b) C_h; (c) C_d; (d) C_t.

following. The value of the four parameters are increased by −50%, −25%, +25%, and +50%, respectively. Tables 5–8 summarize the total costs of WS, HN, EEV, EVPI and VSS, GAP$_{\text{EVPI}}$, and GAP$_{\text{VSS}}$ obtained from all tests. Figure 1 shows the sensitivity analysis on the results for the four parameters. Under current settings, the abovementioned four parameters have a significant impact on the solutions. From Tables 5–8, consistent with theoretical results, it can be founded that WS < HN < EEV in all tests. We can see that the average GAP$_{\text{VSS}}$ are all greater than the average GAP$_{\text{EVPI}}$. The GAP$_{\text{EVPI}}$ of the stochastic model in all tests is nonnegative, indicating the problem of the BRPSD is stochastic by nature. Besides, the GAP$_{\text{VSS}}$ in all tests is nonnegative for the BRPSD, indicating that the problem in the study is suitable for the stochastic model.

From Figure 1, we can see that the increase of C_h, C_d, and C_t rends the increase of HN. We also see that the increase of Q rends the decrease of HN. The parameter that has the greatest impact on HN is the unit transportation cost C_t. This is likely because the contribution of transportation cost to the total objective cost is greater than the penalty cost and the holding cost.

From Figure 1, we can see that, as Q increases, GAP$_{\text{EVPI}}$ tends to decrease, but GAP$_{\text{VSS}}$ increases. It is also noticed that GAP$_{\text{EVPI}}$ and GAP$_{\text{VSS}}$ tend to increase when C_h and C_d increase. However, the value of GAP$_{\text{EVPI}}$ and GAP$_{\text{VSS}}$ contradicts the increase of C_t. The influence of the change of the abovementioned four parameters on GAP$_{\text{VSS}}$ is significantly greater than that of GAP$_{\text{EVPI}}$.

The results illustrate that the GAP$_{\text{EVPI}}$ increases when C_h and C_d increase and Q and C_t decrease, which means that the decision makers are willing to pay more to get accurate redistribution demand information about the future. GAP$_{\text{VSS}}$ increases by increasing the values of C_h, C_d, and Q or decreasing the values of C_t, and consequently, the decision makers can obtain more cost saving resulting from the two-stage stochastic model.

6. Conclusion

In this paper, a general scenario-based two-stage programming model is proposed for bike repositioning problems with stochastic redistribution demands. In the proposed model, the first-stage decisions correspond to the routing decisions and the second-stage decisions correspond to the loading and unloading decisions. The model aims to find the best routes and the ideal quantities of loading and unloading at each station and depot in order to minimize the weighted sum of the expected transportation costs, penalty costs, and holding costs. Then, a simulated annealing algorithm is proposed to solve the model. Number examples are performed to evaluate the proposed model and algorithm, followed by a detailed sensitivity analysis that studies how the change of several important parameters affects the performance of the proposed model and solution algorithm. The proposed model and algorithm have important theoretical and practical significance for the BSS operators, which can reduce the operation costs of the BSS and improve user satisfaction. On the contrary, in addition to bike sharing, other sharing facilities in the sharing economy, such as car sharing, also need to be relocated, so this kind of problem has a strong application background.

In the future, we will apply robust optimization model to solve the bike repositioning problem with stochastic demand in the bike sharing system [35]. We also consider environmental issues by adding the carbon emissions to the objective function of the two-stage stochastic programming model [36–38].

Notations

Sets

N: The set of stations
N_0: The set of nodes, including the stations (indexed by i, $i \in N$) and the depot (indexed by 0)
Ξ: Set of *scenarios*, *indexed by* $\xi \in \Xi$

Parameters

C_t: The travel cost per unit travel time
t_{ij}: The travel time from station i to station j
C_d: The penalty cost per bike at the station
C_h: The holding cost per bike at the depot
Q: The truck capacity
M: A very large number
d_i^ξ: The relocation demand at station i in scenario ξ
p^ξ: The occurrence probability of the scenario ξ

Decision Variables

x_{ij}: Binary variable that equals to 1 if the truck travels directly from station i to station j and 0 otherwise
y_{ij}^ξ: The number of bikes on the truck when it travels directly from station i to station j in scenario ξ
$s_i^{\xi+}$: The excess quantities of bikes at station i after relocation operation in scenario ξ
$s_i^{\xi-}$: The slack quantities of bikes at station i after relocation operation in scenario ξ
q_i: Auxiliary variables associated with station i used for subtour elimination constraints.

Data Availability

All data included in 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

The authors acknowledge the National Natural Science Foundation of China (Grant no. 71271220) and the Project of Social Science Achievement Review Committee of Hunan Province (Grant no. XSP20YBZ165).

References

[1] Y. Yao, Y. Zhang, L. Tian, N. Zhou, and Z. Li, "Analysis of network structure of urban bike-sharing systems: a case study based on real-time data of a public bicycle system," *Sustainability*, vol. 11, no. 19, Article ID 5425, 2019.

[2] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: a classification scheme and survey," *Top*, vol. 15, no. 1, pp. 1–31, 2007.

[3] M. Benchimol, P. Benchimol, B. Chappert et al., "Balancing the stations of a self service "bike hire" system," *RAIRO—Operations Research*, vol. 45, no. 1, pp. 37–61, 2011.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[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] 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.

[11] M. Dell'Amico, M. Iori, and S. Novellani, "A destroy and repair algorithm for the bike sharing rebalancing problem," *Computer & Operations Research.*vol. 71, pp. 149–162, 2016.

[12] P.-S. You, "A two-phase heuristic approach to the bike repositioning problem," *Applied Mathematical Modelling*, vol. 73, pp. 651–667, 2019.

[13] D. Chemla, F. Meunier, and R. Wolfler Calvo, "Bike sharing systems: solving the static rebalancing problem," *Discrete Optimization*, vol. 10, no. 2, pp. 120–146, 2013.

[14] 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.

[15] 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.

[16] Y. Liu, W. Y. Szeto, and S. C. Ho, "A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 208–242, 2018.

[17] 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.

[18] L. D. Gaspero, A. Rendl, and T. Urli, "Balancing bike sharing systems with constraint programming," *Constraints*, vol. 21, no. 2, pp. 318–348, 2016.

[19] 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, pp. 758–773, Springer, Berlin, Germany, 2013.

[20] M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Kloimüllner, "PILOT, GRASP, and VNS approaches for the

static balancing of bicycle sharing systems," *Journal of Global Optimization*, vol. 63, no. 3, pp. 597–629, 2015.

[21] Q. Tang, Z. Fu, and M. Qiu, "A bilevel programming model and algorithm for the static bike repositioning problem," *Journal of Advanced Transportation*, vol. 2019, Article ID 8641492, 19 pages, 2019.

[22] 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.

[23] 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.

[24] M. Dell'Amico, M. Iori, S. Novellani, and A. Subramanian, "The bike sharing rebalancing problem with stochastic demands," *Transportation Research Part B: Methodological*, vol. 118, pp. 362–380, 2018.

[25] 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.

[26] S. Yan, J.-R. Lin, Y.-C. Chen, and F.-R. Xie, "Rental bike location and allocation under stochastic demands," *Computers & Industrial Engineering*, vol. 107, pp. 1–11, 2017.

[27] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *Journal of the ACM (JACM)*, vol. 7, no. 4, pp. 326–329, 1960.

[28] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu, "Development of a fuel consumption optimization model for the capacitated vehicle routing problem," *Computers & Operations Research*, vol. 39, no. 7, pp. 1419–1431, 2012.

[29] H. Guo, C. Li, Y. Zhang, C. Zhang, and Y. Wang, "A nonlinear integer programming model for integrated location, inventory, and routing decisions in a closed-loop supply chain," *Complexity*, vol. 2018, Article ID 2726070, 17 pages, 2018.

[30] Y. Zhang, M. Qi, W.-H. Lin, and L. Miao, "A metaheuristic approach to the reliable location routing problem under disruptions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 83, pp. 90–110, 2015.

[31] Y. Shi, T. Boudouh, O. Grunder, and D. Wang, "Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care," *Expert Systems with Applications*, vol. 102, pp. 218–233, 2018.

[32] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Science & Business Media, Berlin, Germany, 2011.

[33] S.-L. Hu, C.-F. Han, and L.-P. Meng, "Stochastic optimization for joint decision making of inventory and procurement in humanitarian relief," *Computers & Industrial Engineering*, vol. 111, pp. 39–49, 2017.

[34] E. Nikzad, M. Bashiri, and F. Oliveira, "Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty," *Computers & Industrial Engineering*, vol. 128, pp. 358–370, 2019.

[35] Y. Liu, H. Lei, Z. Wu, and D. Zhang, "A robust model predictive control approach for post-disaster relief distribution," *Computers & Industrial Engineering*, vol. 135, pp. 1253–1270, 2019.

[36] J. Jiang, D. Zhang, S. Li, and Y. Liu, "Multimodal green logistics network design of urban agglomeration with stochastic demand," *Journal of Advanced Transportation*, vol. 2019, Article ID 4165942, 2019.

[37] S. Li, Z. Wang, X. Wang, D. Zhang, and Y. Liu, "Integrated optimization model of a biomass feedstock delivery problem with carbon emissions constraints and split loads," *Computers & Industrial Engineering*, vol. 137, Article ID 106013, 2019.

[38] 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.