Transportation Research Part B 90 (2016) 263–278

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

A multiple type bike repositioning problem

Yanfeng Li[a,1], W.Y. Szeto[b,c,*], Jiancheng Long[d,2], C.S. Shui[c]

[a] School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China
[b] The University of Hong Kong Shenzhen Institute of Research and Innovation, Shenzhen, China
[c] Department of Civil Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
[d] School of Automation and Transportation Engineering, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Article history:
Received 13 May 2015
Revised 14 May 2016
Accepted 15 May 2016
Available online 8 June 2016

Keywords:
Static bike repositioning problem
Mixed integer programming
Hybrid genetic algorithm
Greedy heuristic

ABSTRACT

This paper investigates a new static bicycle repositioning problem in which multiple types of bikes are considered. Some types of bikes that are in short supply at a station can be substituted by other types, whereas some types of bikes can occupy the spaces of other types in the vehicle during repositioning. These activities provide two new strategies, substitution and occupancy, which are examined in this paper. The problem is formulated as a mixed-integer linear programming problem to minimize the total cost, which consists of the route travel cost, penalties due to unmet demand, and penalties associated with the substitution and occupancy strategies. A combined hybrid genetic algorithm is proposed to solve this problem. This solution algorithm consists of (i) a modified version of a hybrid genetic search with adaptive diversity control to determine routing decisions and (ii) a proposed greedy heuristic to determine the loading and unloading instructions at each visited station and the substitution and occupancy strategies. The results show that the proposed method can provide high-quality solutions with short computing times. Using small examples, this paper also reveals problem properties and repositioning strategies in bike sharing systems with multiple types of bikes.

1. Introduction

With growing awareness of green transportation, bike sharing systems have become an increasingly popular and powerful complement to public transit in cities around the world. Ideally, travelers can rent a bike at any station and return it to a station with vacant lockers. However, this is not always the case, because the bikes are unevenly distributed in the system. This imbalance can be either temporary, as a result of unidirectional daily commutes, or persistent, due to the topography of the stations. To maintain the service quality of the system, it is necessary to use repositioning trucks to rebalance the bikes at different stations by relocating bikes from stations with an excess to stations with a shortage. This problem is known as the bike repositioning problem or bike rebalancing problem (BRP).

* Corresponding author: Tel.: +852-28578552.
E-mail addresses: yanwaa@126.com (Y. Li), ceszeto@hku.hk, ceszeto@gmail.com (W.Y. Szeto), jianchenglong@hfut.edu.cn (J. Long), samshui2004@hotmail.com (C.S. Shui).
[1] Tel.: +86-28-87600822.
[2] Tel.: +86-551-63831101.

http://dx.doi.org/10.1016/j.trb.2016.05.010

The BRP has drawn attention in recent years. Compared with a dynamic BRP in which the usage rate varies over time (e.g., Contardo et al., 2012; Caggiani and Ottomanelli, 2013; Chemla et al., 2013b; Pfrommer et al., 2014; Kloimüllner et al., 2014), most studies have examined a static BRP (SBRP) because the changes in the bike usage rate are negligible during the repositioning period. Some SBRPs (e.g., Rainer-Harbach et al., 2013; Papazek et al., 2014; Alvarez-Valdes et al., 2015; Brinkmann et al., 2015; Salazar-González and Santos-Hernández, 2015) allow the repositioning trucks to visit a station multiple times. The design objectives of SBRPs include minimizing the total travel cost of the repositioning vehicles (e.g., Benchimol et al., 2011; Lin and Chou, 2012; Chemla et al., 2013a; Salazar-González and Santos-Hernández, 2015), minimizing the unmet demand (e.g., Ho and Szeto, 2014), minimizing the sum of the total travel and handling costs (e.g., Erdoğan et al., 2012; Erdoğan et al., 2014), minimizing the sum of the total travel time and total penalties (e.g., Raviv et al., 2013), minimizing the sum of the total travel time and unmet demand (e.g., Szeto et al., 2016), and minimizing the maximum tour length of the vehicles (e.g., Schuijbroek et al., 2013). Most studies have proposed various approaches to handle the unmet demand in an SBRP. Rainer-Harbach et al. (2013) and Papazek et al. (2013) modeled the unmet demand in the objective function as deviations from the target levels. Benchimol et al. (2011) and Chemla et al. (2013a) set perfect balance as a hard constraint. Erdoğan et al. (2012) relaxed the above constraint by allowing the final inventory level at each station to be within a prespecified demand interval. Schuijbroek et al. (2013) shared a similar idea by setting an inventory interval for service level constraints, whereas Nair et al. (2013) converted the constraint into a probabilistic level-of-service constraint due to the consideration of stochastic demand. Raviv et al. (2013) and Ho and Szeto (2014) adopted a convex penalty function to represent the expected number of shortages for bikes or lockers during the next working day. From these studies, we can observe that the unmet demand can be considered in objective functions, constraints, or penalty functions.

Another important observation is that all of these studies have considered SBRPs with only one type of public bike. However, in reality, there are multiple types of bikes (e.g., bikes with one, two, or three seats and those with a child-seat) in some bike sharing systems (e.g., Hangzhou in China; Taipei in Taiwan; Rotterdam, Hague, and Utrecht in Netherlands; London in the United Kingdom; Aichi in Japan) and more commonly in scenic spots (e.g., Xihu in Hangzhou). Because different types of bikes have different sizes, they occupy different amounts of space in the repositioning vehicles, and the vehicles and the bike stations also need to be partitioned to store different types of bikes. Meanwhile, in situations in which no one-seat bikes are available, a single user may choose to use a two-seat bike, thus affecting the supply of two-seat bikes. Moreover, when no space is available for more small bikes in a designated partition in a repositioning vehicle, they can use up the spaces for large bikes. These two considerations should be captured in the real operation and have motivated us to extend the BRP to capture multiple types of bikes.

The proposed SBRP in this paper deals with multiple types of bikes, which is similar to a multicommodity pickup and delivery problem. In this problem, each of a set of different commodities must be transported from the given pickup locations to the given delivery locations (e.g., Hernández-Pérez and Salazar-González, 2009; Rodríguez-Martín and Salazar-González, 2011; Psaraftis, 2011; Hernández-Pérez et al., 2015; Mahmoudi and Zhou, 2016). A variant of this problem is the swapping problem, which was first introduced by Anily and Hassin (1992) and further studied by Chalasani and Motwani (1999), Erdoğan et al. (2010), and others. In this problem, several commodities must be transported by a unit capacity vehicle. Each customer demands a maximum of one unit of a specific commodity and supplies a maximum of one unit of a different commodity.

The proposed problem is also similar to a multicompartment vehicle routing problem (MCVRP), in which each customer may order multiple types of goods and each vehicle is partitioned into more than one compartment with certain capacities. The problem is to assign all customers to routes so that for each type of goods, the total demand of the customers assigned to any route does not exceed the capacity of the reserved compartment. The objective is to minimize the total transportation cost. This kind of problem is commonly encountered in fuel and oil distribution (e.g., Cornillier et al., 2012; Relvas, 2013; Lahyani et al., 2015), food and grocery distribution (e.g., Chajakis and Guignard, 2003), maritime applications (e.g., Christiansen et al., 2011), and waste collection (e.g., Muyldermans and Pang, 2010; Reed et al., 2014).

The proposed problem, however, differs from the aforementioned problems in several ways. (1) For each type of bike, the pickup or delivery location is not given. That is, any station can serve as a source or a destination of bikes. (2) The pick-up or delivery quantity for each type at each station is a decision variable and has an effect on the objective function value. (3) Some types of bikes (e.g., those with one seat) that are scarce at a station can be substituted by others (e.g., those with two seats or a child-seat; i.e., substitution property). (4) Some types of bikes (e.g., those with one seat) can occupy the empty spaces in the compartments for other types (e.g., those with two seats or a child-seat) in the repositioning vehicles (i.e., occupancy property).

To solve the proposed problem, we may consider solution methods used to solve BRPs, including approximation algorithms (e.g., Benchimol et al., 2011), exact methods (e.g., Raviv et al., 2013; Erdoğan et al., 2014; Dell'Amico et al., 2014), and heuristics. The latter includes classical heuristics for vehicle routing problems (VRPs) based on the actual path distance (e.g., Lin and Chou, 2012), tabu search (e.g., Chemla et al., 2013a; Ho and Szeto, 2014), variable neighborhood search (e.g., Rainer-Harbach et al., 2013), a cluster-first route-second heuristic (e.g., Schuijbroek et al., 2013), a multistage heuristic that addresses the routing and assignment of bike repositioning iteratively (e.g., Angeloudis et al., 2014), and a three-step mathematical programming-based heuristic (e.g., Forma et al., 2015). Exact methods can only obtain optimal solutions in very small instances. For large network applications, it is almost impossible to obtain exact solutions efficiently with exact methods. Hence, heuristics are normally used for such applications.

The hybrid genetic search with adaptive diversity control (HGSADC) developed by Vidal et al. (2012) is a state-of-the-art metaheuristic based on the genetic algorithm (GA) framework introduced by Holland (1975), but it includes many advanced features by combining the exploration of GA with efficient local search–based improvement procedures and diversity management mechanisms. This method has been used to successfully solve a variety of the VRPs with multiple attributes such as the multidepot VRP, the periodic VRP, and the multidepot periodic VRP (Vidal et al., 2012, 2013, 2014) and has been modified to solve the pickup and delivery problem (Cherkesly et al., 2015). Hence, we have adopted the HGSADC as the backbone of our proposed solution method. However, the proposed problem involves pickup and drop-off quantity variables and substitution and occupancy strategy variables in addition to routing variables. Hence, we cannot apply the HGSADC directly to solve our proposed problem, and a new method must be incorporated into the HGSADC to handle the extra complexity. For this purpose, we developed a greedy heuristic to deal with the additional complexity. This heuristic is integrated into the HGSADC to form the proposed combined hybrid GA. To improve the solution quality, more local search operators are also incorporated into the HGSADC. The crossover operator is also modified to suit our application. To show the efficiency and accuracy of the proposed solution method, we set up different test scenarios and compare the results obtained from the exact method. Small examples are also set up to illustrate the problem properties.

The contributions of this study include the following.

- We propose a new SBRP problem with multiple types of bikes, multiple vehicle compartments, and substitution and occupancy properties.
- We examine problem properties.
- We develop an efficient heuristic that can obtain high-quality solutions for large instances.

The remainder of this paper is organized as follows. Section 2 describes and formulates the proposed problem. Section 3 presents the combined hybrid GA. Section 4 depicts the numerical examples. Finally, section 5 gives our conclusions and directions for future research.

2. Problem description and formulation

2.1. Problem description

Consider a complete directed graph $G=(N, A)$ with n_K ($n_K > 1$) types of bikes, where N is the set of nodes and A is the set of arcs. Each node with a positive node number represents a station and the node number 0 represents the depot. Each station is equipped with n_K types of lockers. Each type of bike can only be parked in its own type of locker. The demand for each type of bike is assumed to be known and can be estimated from daily usage. Without loss of generality, stations may have a surplus, a shortage, or just enough bikes. Each station is allowed to have a surplus of certain types and a shortage of others. The depot does not contain bikes. This assumption conforms to the case in which the repositioning operation is performed by a third-party logistics company with no bikes at its company location.

A truck is needed to reposition bikes from stations with a surplus of bikes to those with a shortage to allow more people to use them. The vehicle has n_K separate compartments to accommodate n_K types of bikes, with one compartment specific for one type. Each compartment has a fixed number of spaces. We only consider one vehicle because each district is usually covered by a single truck to redistribute the bikes (Chemla et al., 2013a). The vehicle starts from the depot and returns to the depot after visiting some or all of the stations each night and visits each station no more than once. This implies that it is not necessary for the vehicle to visit all nodes, as in the selective pickup and delivery problem (see for example Ho and Szeto, 2016). The handling cost at each station is considered to be a constant (and not dependent on the loading and unloading quantities) due to the assumption that all bikes are loaded or unloaded simultaneously (i.e., all bikes are loaded or unloaded in one single batch). This constant is incorporated into the travel cost of each link and hence is not explicitly shown in the formulation proposed. A station is allowed to be a pick-up station for one type of bike and a drop-off station for another type. No loading or unloading activities are conducted at the depot.

The company may implement two new repositioning strategies. The first is the *substitution* strategy, in which a shortage of some types of bikes can be solved by providing a specified different type as a substitute. For example, if no one-seat bikes are available at a station, a user can use a two-seat bike as a substitute; the opposite, of course, is infeasible. However, a penalty is associated with each type of substitution to account for a potential reduction of satisfaction of the demand for the specified type at another station. In the previous example, when a two-seat bike is ridden by a single user, one seat of the bike is empty, but that bike can be transported to another station for a pair of users to ride.

The second strategy is the *occupancy* strategy, in which one or more types are stored in the compartments for another designated type during repositioning. For example, when the compartment for one-seat bikes is full, a one-seat bike can be put into an empty spot in the compartment for a two-seat bike (because the unit space of a two-seat bike is larger than that of a one-seat bike), whereas the opposite is infeasible. Again, a penalty is associated with each smaller bike that is put into a space for a larger bike, because some of the space is wasted and because that space cannot be used by a larger bike to be picked up at the next station.

The repositioning problem is to determine the vehicle route; the number of bikes of each type to be loaded, unloaded, and used as substitutes for other types at each visited station; and the number of bikes of each type put into the compartments for their own and other designated types along each arc of the route such that the sum of the imbalance, substitution, and occupancy penalties and the route travel cost is minimized, where an imbalance penalty associated with each type of bike is the monetary value of the shortage or excess of a bike of that type.

2.2. Formulation

The problem is formulated using the following notation.

Sets/Indices

K: Set of types of bikes or rooms in the vehicle, $\{1, \ldots, n_K\}$;
N_c: Set of stations, indexed by $1, \ldots, |N_c|$;
N: Set of nodes, including the stations and the depot, $i = 0, 1, \ldots, |N_c|$.

Parameters

T: The repositioning budget, which is defined as the product of the value of time and the maximum operation duration for repositioning;
p_i^k: Existing number of type k bikes at station i before the repositioning operation starts;
q_i^k: Demand for type k bikes or the target level that induces the minimum penalty at station i;
C_i^k: Number of lockers for type k bikes at station i, (i.e., station capacity for type k bikes);
Q^k: Vehicle capacity for type k bikes;
c_{ij}: Travel cost from node i to node j;
up_i^k: Unit penalty associated with a shortage or excess of one type k bike at station i;
$vp_i^{m,k}$: Unit penalty for a type k bike as a substitute for a type m bike at station i (and $vp_i^{k,k} = 0$);
$op^{m,k}$: Unit penalty for occupying a type m space with a type k bike (and $op^{k,k} = 0$);
$e^{m,k}$: $= 1$ if a type k bike can substitute for a type m bike (and $e^{k,k} = 1$); $= 0$ otherwise;
$f^{m,k}$: $= 1$ if a type k bike can occupy an empty space for type m bikes (and $f^{k,k} = 1$); $= 0$ otherwise;
M_1: A large positive number;
M_2: An upper bound on the number of arcs in the vehicle tour (e.g., $M_2 = |N|$).

Decision variables

x_{ij}: $= 1$ if the vehicle travels directly from node i to node j; $= 0$ otherwise;
$l_{ij}^{m,k}$: Number of type k bikes in the type m compartment when the vehicle travels directly from node i to node j;
y_i^k: Number of type k bikes loaded or unloaded at node i; a *positive* value means *loading* onto the vehicle whereas a *negative* value means *unloading* from the vehicle;
$y_i^{m,k}$: Number of type k bikes loaded into or unloaded from the type m compartment of the vehicle at node i;
$s_i^{m,k}$: Number of type k bikes provided as substitutes for type m bikes at node i; $s_i^{k,k}$ indicates the number of type k bikes provided at node i;
u_i: Auxiliary continuous variable used by the sub-tour elimination constraint;
b_i^k: Auxiliary continuous variable used to linearize $|q_i^k - \sum_{m\in K} s_i^{k,m}|$ in the original objective function.

Formulation

$$\text{Min} \quad \sum_{i\in N}\sum_{j\in N,\ j\neq i} c_{ij}x_{ij} + \sum_{i\in N_c}\sum_{k\in K} up_i^k \cdot b_i^k + \sum_{i\in N_c}\sum_{k\in K}\sum_{m\in K} vp_i^{m,k}\cdot s_i^{m,k} + \sum_{i\in N}\sum_{j\in N,\ j\neq i}\sum_{m\in K}\sum_{k\in K} op^{m,k}\cdot l_{ij}^{m,k}\ . \tag{1}$$

s.t.

Auxiliary constraints:

$$b_i^k \geq q_i^k - \sum_{m\in K} s_i^{k,m}, \quad \forall i \in N_c, \quad \forall k \in K, \tag{2}$$

$$b_i^k \geq \sum_{m\in K} s_i^{k,m} - q_i^k \quad \forall i \in N_c, \quad \forall k \in K. \tag{3}$$

Loading constraints:

$$\sum_{j\in N, j\neq i} l_{ji}^{m,k} + y_i^{m,k} = \sum_{j'\in N, j'\neq i} l_{ij'}^{m,k}, \quad \forall i \in N_c,\ \forall m, k \in K, \tag{4}$$

$$\sum_{k\in K} l_{ij}^{m,k} \leq Q^m x_{ij}, \quad \forall (i, j) \in A,\ \forall m \in K, \tag{5}$$

$$0 \leq l_{ij}^{m,k} \leq f^{m,k}\cdot Q^m, \forall (i, j) \in A,\ \forall m, k \in K, \tag{6}$$

$$y_i^k = \sum_{m\in K} y_i^{m,k}, \quad \forall i \in N_c, \forall k \in K, \tag{7}$$

$$-\left(C_i^k - p_i^k\right) \sum_{j\in N,\, j\neq i} x_{ij} \le y_i^k \le p_i^k \sum_{j\in N,\, j\neq i} x_{ij}, \quad \forall i \in N_c, \forall k \in K, \tag{8}$$

$$-\left(C_i^k - p_i^k\right) \sum_{j\in N,\, j\neq i} x_{ij} \le y_i^{m,k} \le p_i^k \sum_{j\in N,\; j\neq i} x_{ij}, \quad \forall i \in N_c, \forall m, k \in K, \tag{9}$$

$$\sum_{i\in N_c} y_i^k = 0, \quad \forall k \in K, \tag{10}$$

$$\sum_{m\in K} s_i^{m,k} = \left(p_i^k - y_i^k\right), \quad \forall i \in N_c, \forall k \in K, \tag{11}$$

$$0 \le \; s_i^{m,k} \le \; e^{m,k} \cdot M_1, \forall i \in N_c, \; \forall m, k \in K, \tag{12}$$

$$l_{i,0}^{m,k} = 0, \quad \forall (i, 0) \in A, \; \forall m, k \in K, \tag{13}$$

$$l_{0,i}^{m,k} = 0, \quad \forall (0, i) \in A, \; \forall m, k \in K, \tag{14}$$

$$\sum_{i\in N} \sum_{j\in N, j\neq i} c_{ij} x_{ij} \le T, \tag{15}$$

$$y_0^{m,k} = 0, \quad \forall m, k \in K. \tag{16}$$

Routing constraints:

$$\sum_{j'\in N, j'\neq i} x_{ij'} = \sum_{j\in N, j\neq i} x_{ji}, \quad \forall i \in N, \tag{17}$$

$$\sum_{j\in N, j\neq i} x_{ij} \le 1, \quad \forall i \in N_c, \tag{18}$$

$$u_j \ge u_i + 1 - M_2\left(1 - x_{ij}\right), \quad \forall i \in N, \; \forall j \in N_c, \; i \neq j. \tag{19}$$

Integer and definitional constraints:

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N, \quad i \neq j, \tag{20}$$

$$y_i^{m,k}, \quad s_i^{m,k} \text{ Integers}, \quad \forall i \in N, \quad \forall m, k \in K, \tag{21}$$

$$l_{ij}^{m,k} \cdot \text{Integers}, \quad \forall (i, j) \in A, \; \forall m, k \in K, \tag{22}$$

$$u_i \ge 0, \quad \forall i \in N. \tag{23}$$

The objective function (1) minimizes the total cost. The first through fourth terms are the vehicle travel cost, the total imbalance penalties for all types of bikes, the total substitution penalty, and the total occupancy penalty, respectively. Constraints (2) and (3) are the auxiliary constraints used to linearize the unbalanced number of bikes at each station $|q_i^k - \sum_{m\in K} s_i^{k,m}|$.

Constraint (4) is the bike flow conservation condition at a station; it states that for each type of bike, the quantity unloaded from or loaded into a compartment of the vehicle at a station equals the difference between the quantity in the compartment before and after visiting that station. Constraint (5) guarantees that if the vehicle travels directly from node i to node j, the total bike load in each compartment cannot be greater than the corresponding capacity, and equals zero otherwise. Constraint (6) depicts the restrictions of the occupancy strategy. If a bike of a particular type cannot be kept in a compartment intended for another type, then the corresponding load on the vehicle must equal zero; otherwise, the load cannot exceed the compartment capacity. Constraint (7) ensures that the pickup and drop-off quantities for each type of bike at a station equal the total quantities loaded into and unloaded from the different compartments of the vehicle, respectively. Constraint (8) assures that the pickup and drop-off quantities of each type of bike from a visited station cannot exceed the number of bikes of the corresponding type and the number of corresponding empty lockers at the station, respectively.

Constraint (8) also assures that the quantities equal zero if a station is unvisited. Constraint (9) is similar to constraint (8) but applies to the quantity unloaded from or loaded into a compartment of the vehicle at a station. Constraint (10) stipulates that all bikes that are loaded onto the vehicle are unloaded eventually. Constraint (11) guarantees that for each type of bike at each station, the quantity available at the end of repositioning equals the sum of the quantities provided for its own type of demand and the total number of substitutes for other types. Constraint (12) is the restriction for the substitution strategy. If a bike of a particular type is not allowed to be a substitute for another type, then the corresponding number of substitutes must equal zero; otherwise, the number of substitutes is non-negative. Constraints (13) and (14) ensure that the vehicle carries no load to and from the depot, respectively. Constraint (15) limits the vehicle travel cost not to be greater than the repositioning budget.

Constraint (16) ensures that no bike of any type is loaded or unloaded at the depot. Constraint (17) is the vehicle flow-conservation constraint that ensures that if the vehicle visits a station, it must leave that station. Constraint (18) guarantees that no more than one vehicle leaves each node. Constraint (19) is the sub-tour elimination constraint. Constraints (20) to (22) are binary and general integrality constraints for the decisional variables. Constraint (23) is the non-negativity constraint for each auxiliary variable associated with the sub-tour elimination constraint.

3. The solution method

This study develops the combined hybrid GA to solve the proposed problem. This method uses a hybrid GA as the main algorithm to determine the vehicle route and embeds a proposed greedy method to determine the loading or unloading, substitution, and occupancy strategies based on the given route. The hybrid GA and the greedy method are presented in Sections 3.1 and 3.2, respectively.

3.1. The hybrid GA

The hybrid GA is based mainly on the HGSADC developed by Vidal et al. (2012) and modified by Vidal et al. (2013, 2014) and Cherkesly et al. (2015) for the solution of different VRPs. The procedure of the hybrid GA is given below.

Algorithm:
Initialize population
while *the number of iterations without improvement* $< It_{NI}$, *and time* $< T_{max}$, **do**
  Select parents P_1 and P_2
  Generate offspring C_1 and C_2 from P_1 and P_2 (ordered crossover)
  Educate offspring C_1 and C_2 (local search procedure)
  if $C_i,\ i = 1,\ 2$ is infeasible, **then** insert it into infeasible subpopulation; repair with probability P_{repair}
  if $C_i,\ i = 1,\ 2$ is feasible, **then** insert it into feasible subpopulation
  if *maximum subpopulation size is reached*, **then** select survivors
  if *the best solution is not improved for* It_{div} *iterations*, **then** diversify population
  Adjust the penalty parameters for violating feasibility conditions
end while
Return the best feasible solution

The algorithm structure is the same as that given by Vidal et al. (2012, 2013, 2014). The key differences between the HGSADC and the hybrid GA involve solution representation, education, the crossover operator, and the inclusion of the embedded greedy method for fitness calculation to determine high-quality solutions to the studied problem.

3.1.1. Solution representation

An individual, route, or solution P is represented by a permutation of visited stations. The length of each solution L equals the number of visited stations. For example, there are 9 visited stations on a vehicle route. A sample solution can be represented as $2-3-7-4-5-1-9-6-8$, where stations 2 and 8 are the first and last stations, respectively. The depot is not captured in any solution.

3.1.2. Evaluation of individuals

Each solution P gives an original objective function value of the problem $f(P)$. $f(P)$ is the sum of the route travel costs and the total imbalance, substitution, and occupancy penalties. The route travel cost $R(P)$ equals $\sum_{h=0}^{L} c_{i_h,i_{h+1}}$, where i_h is the h-th visited station, $h = 1,\ \ldots, L$ $(L \le |N_c|)$, and $i_0 = i_{L+1} = 0$ (i.e., the depot). The total imbalance penalty equals $\sum_{i \in N_c} \sum_{k \in K} up_i^k \cdot |q_i^k - \sum_{m \in K} s_i^{k,m}\ |$. The total substitution and occupancy penalties are respectively defined by the third and the last terms of (1). The three total penalties are functions of either $[l_{ij}^{m,k}]$ or $[s_i^{m,k}]$, both of which are determined by the greedy heuristic.

To consider solution infeasibility, $f(P)$ is modified to the fitness value, $Z(P)$, defined as

$$Z(P) = f(P) + \alpha w(P) + M\delta, \tag{24}$$

where $w(P) = \max\{0,\ R(P) - T\}$, α represents the penalty parameter associated with $w(P)$, and M is a large positive number. δ equals 1 if any bikes are finally sent to the depot, and 0 otherwise.

Position	1	2	3	4	5	6	7	8	9
			$t = 3$ ↓			$y = 6$ ↓			
P_1	1	4	|6	9	7	5|	3		
P_2	3	8	|5	6	2	7|	9	1	4
C_1	8	2	6	9	7	5	1	4	3
C_2	1	4	5	6	2	7	3		

Fig. 1. Example of the modified OX crossover.

To consider diversity in each subpopulation, the fitness value is further modified to a biased fitness value $BF(P)$, which is defined as

$$BF(P) = fit(P) + \left(1 - \frac{N_{elite}}{N_{indiv}}\right) dc(P), \tag{25}$$

where N_{elite} is the number of elite individuals that survive to the next generation and N_{indiv} is the actual number of individuals in the subpopulation. $fit\ (P)$ and $dc\ (P)$ are the ranks of a solution P in a subpopulation with respect to $Z(P)$ and its similarity with the other solutions. As in the study of Cherkesly et al. (2015), the similarity of two solutions is measured by the number of common arcs. The similarity of one solution with all other solutions is defined as the sum of the number of common arcs between that solution and every other solution.

3.1.3. Parent selection and crossover

As in Vidal et al. (2012, 2013, 2014) and Cherkesly et al. (2015), each of two parents is selected from a binary tournament, which randomly picks two individuals from the entire population and retains the one with the best biased fitness. The two selected parents are used to generate two children based on the simple ordered crossover (OX; Prins, 2004; Vidal et al., 2014). However, the solution length varies from one to another. Therefore, OX is modified to ensure that the break-points are smaller than the length of the shorter solution. Fig. 1 shows an example of how to construct two children with the modified OX crossover operator. Note that the length of C_1 is the same as that of P_2.

The following procedure describes the offspring creation:

Step 1: Select two break-points t and y randomly with $t < y \le \min\{L_{P_1}, L_{P_2}\}$, where L_{P_1} and L_{P_2} are the lengths of P_1 and P_2, respectively.
Step 2: Copy the sub-tour $(P_1(t), \ldots, P_1(y))$ from P_1 to its corresponding position in the child C_1.
Step 3: Sweep P_2 circularly from $y + 1$ to fill unvisited nodes in C_1 circularly from $y + 1$.
Step 4: Obtain C_2 by exchanging the roles of P_1 and P_2.

The whole procedure can be implemented in $O(\max\{L_{P_1}, L_{P_2}\})$.

3.1.4. Education and repairing

The education operator is adopted with the probability P_m to improve the offspring solution quality. It is formed by one or more of the following nine moves.

1. *Single node exchange*: The operator randomly selects and swaps two visited stations.
2. *Multiple node exchanges*: The operator randomly selects and swaps two independent sub-tours with random lengths.
3. *Single node relocation*: The operator randomly selects and relocates a visited node into another random position.
4. *Multiple node relocations*: The operator randomly selects and relocates a sub-tour of the route into another random position.
5. *2-opt:* The operator randomly removes a pair of non-consecutive arcs from the route and reinserts two new arcs to form a route.
6. *Single node insertion*: The operator randomly selects and inserts an unvisited node into a random position of the route.
7. *Multiple node insertions*: The operator randomly selects a set of unvisited nodes UN with $|UN| \ge 2$, randomly makes a sequence from these nodes, and inserts the sequence into a random position of the route.
8. *Single node deletion*: The operator randomly selects and deletes a visited station.
9. *Multiple node deletions*: The operator randomly selects and deletes a sub-tour of the route.

The first four intra-route moves are taken from Cherkesly et al. (2015), and the others are newly added intra-route moves to solve the problem. The Education procedure follows.

Step 1: Put all nine types of moves into a possible move choice set.
Step 2: Select one type of move from the set.
Step 3: If the selected type can improve the solution in one move, the same type of move is selected repeatedly until no improvement is found in IT_{edu} consecutive moves and the *Education* phase ends.

Step 4: Remove the selected type of move from the possible move choice set.
Step 5: If no type of move is in the possible move choice set, stop Education. Otherwise, go to Step 2.

The solutions from the *Education* operator may be feasible and infeasible according to the service duration. They are put into the corresponding sub-population. Infeasible educated individuals must undergo the *Repair* operation with probability P_{Repair}. If *Repair* is successful, the resulting individual is added to the feasible subpopulation (but the infeasible individual is not deleted from the infeasible subpopulation).

Repair consists of temporarily multiplying the penalty parameter α by 10 and restarting the *Education* operation. When the resulting individual is still infeasible, α is temporarily multiplied by 100 and the *Education* operation is started again. This significant increase in the penalty parameter aims to redirect the search toward feasible solutions.

3.1.5. Population management and search guidance

The size of two sub-populations is controlled within the range $[\mu,\ \mu+\lambda]$, where μ is the minimum subpopulation size and λ is the number of offspring in a generation. Any new individual generated by *Crossover, Education,* and *Repair* operations is directly added to the appropriate subpopulation with respect to its feasibility. If the subpopulation number reaches its maximum size $\mu+\lambda$, *survivor selection* is implemented by eliminating the individuals with the λ highest biased fitness values until the subpopulation size decreases to μ.

In the beginning of the hybrid GA, 4μ individuals are generated to form the population input to the reproduction process (Vidal et al., 2012, 2013, 2014). Each initial individual is created by randomly choosing and adding stations to the route one by one until all stations are added to the route. These initial individuals then undergo *Education* with a probability of 1. If an educated individual is infeasible, it will undergo *Repair* with a probability of 0.5. Educated or repaired individuals are put into the corresponding subpopulations in terms of feasibility. *Survivor selection* is activated when a subpopulation reaches the maximum size $\mu\ +\ \lambda$.

The penalty parameter α for infeasible individuals is dynamically adjusted during the algorithm to balance the proportions of the feasible and infeasible individuals. The *adjustment* is done every 100 iterations. Let ξ^{REF} be the target proportion of feasible individuals. If the proportion of feasible individuals with respect to $w(P)$ is less than $\xi^{REF}-5\%$ or greater than $\xi^{REF}+5\%$, then α is adjusted by multiplying it by 1.2 or 0.85, respectively.

To regularly introduce new genetic material, the *diversification* operation is triggered after each It_{div} successive iterations without improvement of the best solution. It_{div} is usually set as $0.4It_{NI}$. Diversification consists of retaining the best $\mu/3$ individuals of each subpopulation, creating 4μ new individuals, and replacing the others with survivor operators.

3.2. The embedded greedy heuristic

For a given route generated by the hybrid GA, the embedded method determines $[y_i^k]$, $[y_i^{m,k}]$, $[l_{ij}^{m,k}]$, and $[s_i^{m,k}]$, where the last two vectors are used to determine a route's fitness value. This method builds solutions following a local best successor strategy. Because there are no bikes at the depot, $l_{0i_1}^{m,k}=y_0^{m,k}=0$. Based on this initial condition, the elements of $[y_i^k]$, $[y_i^{m,k}]$, $[l_{ij}^{m,k}]$, and $[s_i^{m,k}]$ are determined following the visit order of the stations. The overall procedure is depicted as follows:

Step 0: Set $l_{0i_1}^{m,k}=y_0^{m,k}=0,\ \forall m,\ k\in K;\ h\ =\ 1.$
Step 1: Determine $y_{i_h}^k,\ \forall k\in K.$
Step 2: Determine $y_{i_h}^{m,k},\ \forall m,k\in K.$
Step 3: Determine $l_{i_h,i_{h+1}}^{m,k},\ \forall m,k\in K.$
Step 4: Determine $s_{i_h}^{m,k},\ \forall m,k\in K.$
Step 5: If $h\ =\ L$, stop. Otherwise, set $h\ =\ h\ +\ 1$; Go to Step 1.

The following subsections depict Steps 1 through 4 in details.

3.2.1. Step 1: determine $y_{i_h}^k, \forall k\in K$

If $1\le h<L$, then at i_h,

$$y_{i_h}^k=\begin{cases}-\min\left(q_{i_h}^k-\ p_{i_h}^k,\sum\limits_{m\in K}l_{i_{h-1},i_h}^{m,k}\right) & \text{if } q_{i_h}^k-\ p_{i_h}^k\ge 0,\ \forall k\in K;\\ \min\left(\ p_{i_h}^k-q_{i_h}^k,\ \sum\limits_{m\in K}f^{m,k}Q^m-\sum\limits_{m\in K}l_{i_{h-1},i_h}^{m,k}\right) & \text{if } q_{i_h}^k-\ p_{i_h}^k<0,\ \forall k\in K.\end{cases} \tag{26}$$

We assume that no bikes are allowed to remain on the vehicle when it returns to the depot and that each station is visited no more than once. Therefore, at the last station ($h=L$), all bikes on the vehicle should be unloaded regardless of the station imbalance. Then,

$$y_{i_L}^k = -\min\left(C_{i_L}^k - p_{i_L}^k, \sum_{m\in K} l_{i_{L-1},i_L}^{m,k} \right), \ \forall k \in K. \tag{27}$$

Note that if $C_{i_L}^k - p_{i_L}^k < \sum_{m\in K} l_{i_{L-1},i_L}^{m,k}$, then some bikes are left on the vehicle due to a shortage of lockers at the last station of the sequence. In this case, we add a very large penalty M to the fitness function to penalize this worst arrangement because the unloading quantity is infeasible.

3.2.2. Step 2: determine $y_{i_h}^{m,k}, \forall m, k \in K$

We determine $y_{i_h}^{m,k}, \ \forall m, k \in K$ from $y_{i_h}^k, \ \forall k \in K$ based on the occupancy penalty saving principle. Any bikes picked up from (or delivered to) a station should be placed in (or obtained from) the compartment with a unit occupancy penalty as small as (or as large as) possible. To achieve this, we define and initialize an occupancy priority value as follows:

$$opr^{m,k} = 1/op^{m,k} \cdot f^{m,k}, \forall m, k \in K. \tag{28}$$

According to the above, if the unit occupancy penalty $op^{m,k}$ is higher, then the occupancy priority value is smaller, meaning that the priority of putting type k bikes into a type m compartment is lower and the priority of getting a type k bike from that compartment is higher. Note that $opr^{m,k}$ is 0 if type k bikes cannot occupy the compartment for type m bikes, and $opr^{k,k}$ is positive infinity.

$y_{i_h}^{m,k}, \ \forall m, k \in K$ at station i_h is determined by the following procedure:

Step 2.0: Set $l'^{m,k}_{i_{h-1},i_h} = l^{m,k}_{i_{h-1},i_h}, \ \forall m, k \in K$, and $opr^{m,k}, \forall m, k \in K$ according to (28).
Step 2.1: If $y_{i_h}^k \geq 0, \forall k \in K$, go to Step 2.5.
Step 2.2: Select $g, d \in K$ such that $opr^{g,d}$ is the least positive.
Step 2.3: $y_{i_h}^{g,d} = -\min(l'^{g,d}_{i_{h-1},i_h}, \ -y_{i_h}^d)$.
Step 2.4: Update $l'^{g,d}_{i_{h-1},i_h} = l'^{g,d}_{i_{h-1},i_h} + y_{i_h}^{g,d}$, $y_{i_h}^d = y_{i_h}^d - y_{i_h}^{g,d}$, and $opr^{g,d} = 0$. Go to Step 2.1.
Step 2.5: If $y_{i_h}^k = 0, \forall k \in K$, stop.
Step 2.6: Choose $g, d \in K$ such that $opr^{g,d}$ is the largest.
Step 2.7: Calculate $y_{i_h}^{g,d} = \min(Q^g - \sum_{k\in K} l'^{g,d}_{i_{h-1},i_h}, \ y_{i_h}^d)$.
Step 2.8: Update $l'^{g,d}_{i_{h-1},i_h} = l'^{g,d}_{i_{h-1},i_h} + y_{i_h}^{g,d}$, $y_{i_h}^d = y_{i_h}^d - y_{i_h}^{g,d}$, and $opr^{g,d} = 0$. Go to Step 2.5.

3.2.3. Step 3: determine $l_{i_h,i_{h+1}}^{m,k}, \forall m, k \in K$

After $y_{i_h}^{m,k}, \ \forall m, k \in K$ is known from Step 2, $l_{i_h,i_{h+1}}^{m,k}, \ \forall m, k \in K$ is determined by $l_{i_h,i_{h+1}}^{m,k} = l_{i_{h-1},i_h}^{m,k} + y_{i_h}^{m,k}, \ \forall m, k \in K$.

3.2.4. Step 4: determine $s_{i_h}^{m,k}, \forall m, k \in K$

To obtain substitution quantities, we set a substitution priority value for each station based on the penalty saving principle. At station i_h, each substitution priority value $spr_{i_h}^{m,k}$ is initialized by

$$spr_{i_h}^{m,k} = \left(up_{i_h}^m - up_{i_h}^k - vp_{i_h}^{m,k}\right) \cdot e^{m,k}, \forall m, k \in K. \tag{29}$$

If $spr_{i_h}^{m,k}$ is positive, it means that the substitution of a type k bike for a type m bike decreases the total penalty, including the imbalance and substitution penalties at station i_h. If the value is negative, the substitution increases the total penalty. If a type k bike cannot be substituted for a type m bike at station i_h, then $spr_{i_h}^{m,k} = 0$. Note that $spr_{i_h}^{k,k} = 0$. A higher substitution priority value brings a greater penalty reduction, and the substitution strategy should therefore be performed with the highest positive value first.

The substitution quantities $s_{i_h}^{m,k}, \ \forall m, k \in K$ are then calculated by the following procedure:

Step 4.0: Set $p'^k_{i_h} = p_{i_h}^k - y_{i_h}^k, \ \forall k \in K$, and $spr_{i_h}^{m,k}, \ \forall m, k \in K$ according to (29).
Step 4.1: If $spr_{i_h}^{m,k} \leq 0, \ \forall m, k \in K$ or $q_i^k - p'^k_{i_h} \leq 0, \ \forall k \in K$, stop.
Step 4.2: Select $g, d \in K, g \neq d$ such that $spr_{i_h}^{g,d}$ is the largest and $q_i^g - p'^g_{i_h} > 0$.
Step 4.3: Compute $s_{i_h}^{g,d} = \min(q_{i_h}^g - p'^g_{i_h}, \ p'^d_{i_h})$.
Step 4.4: Set $p'^g_{i_h} = p'^g_{i_h} + s_{i_h}^{g,d}$, $p'^d_{i_h} = p'^d_{i_h} - s_{i_h}^{g,d}$, and $spr_{i_h}^{g,d} = 0$. Go to Step 4.1.

Fig. 2. Illustration of the network with two stations.

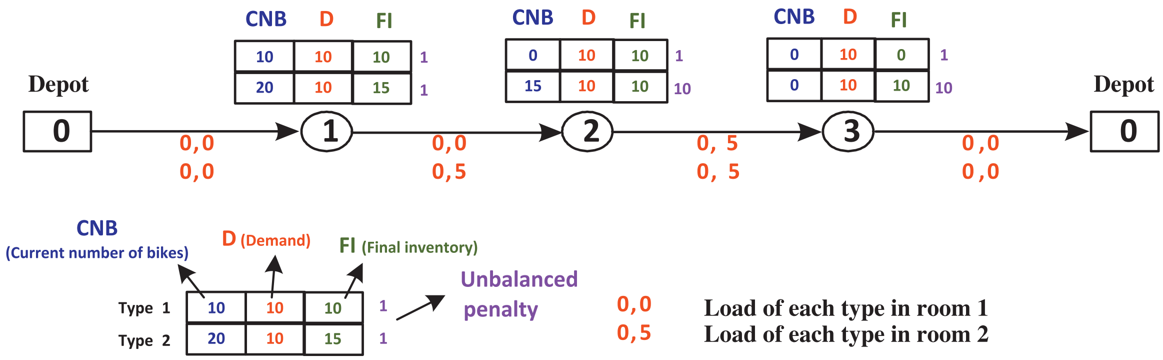

Fig. 3. Illustration of scenario 5 with three stations.

4. Numerical studies

In this section, numerical examples are set up to illustrate the problem properties and the performance of the combined hybrid GA method. The solution method was coded in C++ on an Intel Xeon CPU E5-1620 @3.7 GHz with 32 GB of RAM and used to solve examples in Sections 4.4 and 4.5. The examples in Sections 4.1 through 4.4 were solved with the branch and cut method in IBM-ILOG CPLEX 12.6.1. All examples only consider two types of bikes, type 1 and type 2. Type 2 bikes may substitute for type 1 bikes if substitution is allowed, and type 2 bikes may occupy the compartment for type 1 bikes in the vehicle if the occupancy strategy is allowed, but not vice versa for either strategy.

4.1. Problem properties and various best operational strategies

This section makes use of two small networks shown in Figs. 2 and 3 to show the properties of the multiple-type BRP and the effect of the parameter setting on the best operational strategy. A total of six scenarios are investigated, and all have the same settings:

(1) The travel cost of any arc is 10 to simplify the routing strategy;
(2) The capacity of each type of bike at each station is set as 20;
(3) The repositioning budget T is set as 500;
(4) Both substitution and occupancy strategies are allowed;
(5) The unit penalties for substitution and for occupying a compartment for the other type are both set as 1 ($vp_i^{1,2} = op^{1,2} = 1,\ i = 1,\ 2$).

For Scenarios 0 through 4, Fig. 2 is used, and the demand and the existing inventories are set as:

$$p_1^1 = 0,\ \ p_1^2 = 10,\ p_2^1 = p_2^2 = 0,\ q_1^1 = q_1^2 = q_2^1 = q_2^2 = 10.$$

Scenario 0: Base case, do nothing

We set $up_1^1 = up_1^2 = up_2^1 = up_2^2 = 1,\ \ Q^1 = Q^2 = 10.$

The solution is: $l_{12}^{1,1} = l_{12}^{2,1} = l_{12}^{1,2} = l_{12}^{2,2} = 0,\ s_1^{1,2} = s_2^{1,2} = 0$. In this case, although type 2 bikes can substitute for type 1 bikes at station 1, no substitution of type 2 bikes for type 1 bikes occurs, because the imbalance penalty for type 1 bikes at station 1 is still not sufficiently large such that the substitution of type 2 bikes for type 1 bikes cannot reduce the total cost (i.e., $up_1^1 - up_1^2 - vp_1^{1,2} \leq 0$).

Scenario 1: Substitution within a station

We set $up_1^1 = 10,\ up_1^2 = up_2^1 = up_2^2 = 1,\ Q^1 = Q^2 = 10.$

When up_1^1 is increased to 10, the solution becomes: $l_{12}^{1,1} = l_{12}^{2,1} = l_{12}^{1,2} = l_{12}^{2,2} = 0$, $s_1^{1,2} = 10,\ s_2^{1,2} = 0$. In this case, a comparatively large imbalance penalty for type 1 bikes induces a substitution of type 2 bikes for type 1 bikes to minimize the total cost (because $up_1^1 - up_1^2 - vp_1^{1,2} > 0$). Moreover, no vehicle is used because the imbalance penalties for both types of bikes at station 2 are smaller than or equal to the corresponding penalties at station 1, and the repositioning activity does not reduce the total cost.

Scenario 2: Occurrence of simple repositioning activity

We set $up_1^1 = up_1^2 = 1, up_2^1 = up_2^2 = 5,\ Q^1 = Q^2 = 10.$

When up_2^1 and up_2^2 are both increased to 5, the solution becomes: $l_{12}^{1,1} = l_{12}^{2,1} = l_{12}^{1,2} = 0,\ l_{12}^{2,2} = 10,\ s_1^{1,2} = s_2^{1,2} = 0$. With larger imbalance penalties at station 2 for both types of bikes, the vehicle picks up all type 2 bikes at station 1 and delivers them to station 2 without using the substitution or occupancy strategies because implementation of these strategies cannot reduce the total cost.

Scenario 3: Implementing an occupancy and repositioning strategy

We set $up_1^1 = up_1^2 = 1,\ up_2^1 = up_2^2 = 5,\ Q^1 = 10,\ Q^2 = 5.$

When up_2^1 and up_2^2 are both increased to 5 and the capacity for type 2 bikes is reduced to 5, the solution becomes: $l_{12}^{1,1} = l_{12}^{1,2} = 0,\ l_{12}^{1,2} = l_{12}^{2,2} = 5,\ s_1^{1,2} = s_2^{1,2} = 0$. In this case, the vehicle capacity for type 2 bikes is not sufficient to accommodate the required number of bikes for station 2. Hence, the type 1 bike compartment is occupied by type 2 bikes (i.e., the occupancy strategy is implemented), but substitution does not occur at station 2.

Scenario 4: Substitution across different stations (waiting strategy for substitution)

We set $up_1^1 = up_1^2 = 1,\ up_2^1 = 10,\ up_2^2 = 1,\ Q^1 = Q^2 = 10.$

When up_2^1 is increased to 10, the solution becomes: $l_{12}^{1,1} = l_{12}^{2,1} = l_{12}^{1,2} = 0,\ l_{12}^{2,2} = 10,\ s_1^{1,2} = 0,\ s_2^{1,2} = 10$. In this case, we can observe that the high penalty for type 1 bikes at station 2 induces all type 2 bikes at station 1 to be delivered to station 2 as substitutes for type 1 bikes. This activity can be regarded as the waiting strategy for substitution, which means that some type 2 bikes are substituted for type 1 bikes at the following station(s), instead of at the one currently visited, mainly because of the much higher imbalance penalty at the following station(s).

Scenario 5: Waiting strategy for the occupancy strategy

Scenario 5 uses the setting shown in Fig. 3 to illustrate the waiting strategy for the occupancy strategy, which means that some spaces are left on the vehicle to accommodate the bikes to be loaded at the following visited station(s), and these bikes are placed in the compartment designated for other types of bikes. We set $Q^1 = 10,\ Q^2 = 5$. At station 1, there are 10 redundant type 2 bikes. However, only five type 2 bikes are picked up and placed in compartment 2, instead of occupying the vacancies in compartment 1. Those remaining vacancies are occupied when the vehicle arrives and pick up five type 2 bikes at station 2, where there is a much higher imbalance penalty for type 2 bikes. As a result, the demand for type 2 bikes at stations 2 and 3 are just satisfied, and the total penalty is minimized.

To sum up, depending on the parameter setting, a substitution strategy may be adopted across stations or within a station and a waiting strategy for the occupancy strategy may be applied.

4.2. Effect of substitution and occupancy strategies toward the total cost

This section adopts a five-station network to compare different combinations of substitution and occupancy strategies toward the total cost (i.e., the objective value). The travel cost between any two stations and the station capacity of type

Table 1
Computation results in four strategies.

Different strategies	Objective value	Travel cost	Imbalance penalty	Substitution penalty	Occupancy penalty	CPU time (s)
SA+OA	178.75	138	40	0.5	0.25	0.263
SA+ON	269.40	139	130	0.4	0	0.245
SN+OA	648.25	138	510	0	0.25	0.283
SN+ON	669.00	139	530	0	0	0.220

SA – substitution strategy is allowed; OA – occupancy strategy is allowed; SN – substitution strategy is not allowed; ON – occupancy strategy is not allowed.

1 bikes are derived from B_010_1_00.bbs, which was generated by Rainer-Harbach et al. (2013) and can be obtained from https: //www.ads.tuwien.ac.at/w/Research/Problem_Instances. We choose the first five stations. Initially, no bikes are present at stations 1 and 2 and other stations have 20 type 2 bikes. The station capacity of type 2 bikes is set to be the same as that of type 1 bikes. Other parameter settings are as follows: $Q^1 = 20,\ Q^2 = 5,\ q_i^1 = q_i^2 = 10,\ up_i^1 = 10, up_i^2 = 1, \forall i,\ vp_i^{m,k} = 0.01,\ \forall i, m, k, op^{m,k} = 0.01,\ \forall m, k,\ T = 500$. Table 1 gives the optimal results in four different strategies.

Table 1 shows that the substitution strategy can achieve much lower costs here because of the much higher imbalance penalty for type 1 bikes. Meanwhile, the occupancy strategy can also reduce the total cost, although to a lesser extent. It can allow the vehicle capacity to be used more effectively, leading to more substitutions to satisfy the demand. The total cost is minimized if both substitution and occupancy strategies are allowed, which implies that the agency should adopt both more flexible substitution and occupancy strategies in their operation whenever possible.

4.3. Sensitivity analysis of penalties for imbalance, substitution, and occupation of other spaces

To show the effect of the penalties for imbalance, substitution, and occupation of spaces in compartments for other types toward the final solution, we increase the imbalance penalties for type 1 and type 2 bikes, the substitution penalty, and the occupancy penalty separately and set a base case for comparison in which all penalties equal one. We adopt the five-station network and the settings for the vehicle capacities, demand levels, and maximum operation duration for repositioning in Section 4.2. Substitution and occupancy strategies are both allowed. The combinations of penalty parameters and the corresponding performance measures are shown in Table 2. The percentages in Table 2 are defined as follows:

$$\text{Imbalance percentage of a particular type} = \frac{\text{the sum of imbalance amount of that type at each station}}{\text{the total demand of that type at each station}}$$

$$\text{Substitution percentage} = \frac{\text{the sum of the substitution amount of type 2 bikes for type 1 bikes at each station}}{\text{the total existing amount of type 2 bikes of all stations}}$$

$$\text{Occupancy percentage} = \frac{\text{the sum of the occupancy amount of type 2 bikes in the room of type 1 bikes on each link}}{\text{the number of links} \times \text{vehicle capacity for type 1 bikes}}$$

According to Tables 2a and b, an increase in the imbalance penalty for a specific type of bike can reduce its imbalance percentage in general and even to 0% in an extreme case, but the imbalance percentage can remain unchanged, together with other measures. For example, although the unit imbalance penalty for type 1 bikes increases from 1 to 1.8, the imbalance percentage of type 1 bikes, the imbalance percentage of type 2 bikes, and the substitution percentage do not change, and it is not necessary to adopt the occupancy strategy because, in the range between 1 and 1.8, the sum of the substitution and occupancy penalties is still higher than the sum of the imbalance penalty and the transportation costs. Only a simple repositioning strategy is used to transport five type 2 bikes to station 1 and another 5 to station 2.

Table 2a also shows that when the unit imbalance penalty for type 1 bikes increases from 1.8 to 3, the substitution percentage in group 1 increases in general, and the imbalance percentage of type 1 bikes and the substitution percentage are reduced correspondingly because more type 2 bikes are used as substitutes to satisfy the demand for type 1 bikes. However, it is still not necessary to use the occupancy strategy because the occupancy penalty is higher than the sum of the imbalance and substitution penalties. The occupancy strategy is used only when the unit imbalance penalty for type 1 bikes increases to at least 3.1. In this case, all of the demand for type 1 bikes is satisfied by using both the substitution and occupancy strategies. The occupancy percentage becomes 10%, because five type 2 bikes at station 1 and five type 2 bikes at station 2 are put in a type 1 compartment with a capacity of 20 during transportation.

Table 2b shows that when the unit imbalance penalty of type 2 bikes for each station increases, a higher priority is given to satisfy the type 2 demand, and the number of type 2 bikes used as substitutes for type 1 bikes can decrease. Moreover, when $up_i^2 \geq 1.1$ ($\forall i$), an occupancy strategy is adopted to deal with the shortage of type 2 bikes at stations 1 and 2 because the imbalance penalty of bike 2 types is higher than the sum of the occupancy penalty and the transportation costs, thus leading to an occupancy percentage of 10%.

Table 2c shows that an increase in the unit substitution penalty for each station can decrease the substitution percentage. However, when the unit cost is 2.1 or more, no substitution occurs because all type 2 bikes are used only to satisfy all of

Table 2
Computation results with different combinations of unit penalties.

a) Group 1: Variation in unit imbalance penalty for type 1 bikes

up_i^1 $(\forall i)$	Imbalance percentage of type 1 bikes	Imbalance percentage of type 2 bikes	Substitution percentage	Occupancy percentage	Number of substitutions at each station	Number of type 2 bikes at each station at the end
1–1.8	60%	20%	33.33%	0	0, 0, 5, 5, 10	5, 5, 10, 10, 10
1.9	50%	30%	41.67%	0	0, 0, 10, 5, 10	5, 5, 5, 10, 10
2	40%	40%	50%	0	0, 0, 10, 10, 10	0, 0, 10, 10, 10
2.1–2.2	30%	50%	58.33%	0	0, 5, 10, 10, 10	0, 0, 10, 10, 5
2.3–3	20%	60%	66.67%	0	5, 5, 10, 10, 10	0, 0, 0, 10, 10
3.1–5	0%	80%	83.33%	10%	10, 10, 10, 10, 10	0, 0, 0, 0, 10

b) Group 2: Variation in unit imbalance penalty for type 2 bikes

up_i^2 $(\forall i)$	Imbalance percentage of type 1 bikes	Imbalance percentage of type 2 bikes	Substitution percentage	Occupancy percentage	Number of substitutions at each station	Number of type 2 bikes at each station at the end
1	60%	20%	33.33%	0	0, 0, 5, 5, 10	5, 5, 10, 10, 10
1.1–5	80%	0%	16.67%	10%	0, 0, 0, 5, 5	10, 10, 10, 10, 10

c) Group 3: Variation in unit substitution penalty

$vp_i^{1,2}$ $(\forall i)$	Imbalance percentage of type 1 bikes	Imbalance percentage of type 2 bikes	Substitution percentage	Occupancy percentage	Number of substitutions at each station	Number of type 2 bikes at each station at the end
1	60%	20%	33.33%	0	0, 0, 5, 5, 10	5, 5, 10, 10, 10
1.1–2	80%	0%	16.67%	10%	0, 0, 0, 0, 10	10, 10, 10, 10, 10
2.1–5	100%	20%	0%	10%	0, 0, 0, 0, 0	10, 10, 10, 10, 20

d) Group 4: Variation in unit occupancy penalty

$op^{1,2}$	Imbalance percentage of type 1 bikes	Imbalance percentage of type 2 bikes	Substitution percentage	Occupancy percentage	Number of substitutions at each station	Number of type 2 bikes at each station at the end
1–5	60%	20%	33.33%	0	0, 0, 5, 5, 10	5, 5, 10, 10, 10

the type 2 demand and the substitution cost is too high compared with the total imbalance penalty of the type 1 demand at stations 1 and 2, which prevents the adoption of the occupancy strategy with the remaining 10 type 2 bikes to satisfy the unbalanced type 1 demand. Therefore, the imbalance percentage of type 1 bikes remains 100% and that for type 2 bikes remains 20%.

Table 2d shows that a sole variation in the occupancy penalty for each station does not lead to any changes to the solution obtained because there is no alternative occupancy strategy. If there were more than two types of bikes, more than one occupancy strategy could be chosen, and the unit occupancy parameters would then determine which occupancy strategy should be selected.

To conclude, optimal solutions can be sensitive not only to the unit penalties for imbalance but also to substitution. These penalties should be estimated accurately for practical operations, but this will be left to future studies. Moreover, the substitution and occupancy strategies may not be used simultaneously even when both are allowed.

4.4. Performance analysis of the combined hybrid GA

We adopt the instances of the BRP with a single commodity generated by Rainer-Harbach et al. (2013) with sizes varying from 10 stations to 180 stations. Their data are for type 1 bikes. We set the vehicle capacity for type 2 bikes to five, the capacity of type 2 bikes at each station to be the same as that of type 1 bikes, and the type 2 demand at each station to be half of the corresponding capacity. The existing number of type 2 bikes at each station is randomly generated on the basis of a uniform distribution from 0 to the corresponding station's capacity. The substitution and occupancy strategies are both allowed. The unit imbalance penalties associated with type 1 and type 2 bikes at each station are set at 10 and 1, respectively. Both the unit substitution and the occupancy penalties are set at 1.

4.4.1. Parameter setting

Vidal et al. (2012) performed an extensive meta-calibration experiment to generate good parameter values on several variants of VRP instances. They found that the optimal set of parameters appears to be independent of the problem type except for the number of offspring in a generation λ. Vidal et al. (2013, 2014) adopted the same parameter setting as in Vidal et al. (2012); therefore, we also adopt the same parameter setting in Vidal et al. (2012) in this section. That is, we have $\mu = N_{indiv} = 25$, $N_{elite} = 10$, $P_m = 1.0$, $P_{\ repair} =\ 0.5$, $\xi^{REF} = 0.2$. Because the BRP is the closest to the PVRP when the number of periods is 1, we set the same generation size $\lambda = 40$ for periodic VRP as in Vidal et al. (2012). $\alpha = 2$,

Table 3
Comparison of the performance of the exact method and the combined hybrid GA.

Network size $\|N\|$	Exact method				Combined hybrid GA method			
	CPU (s)	UB	LB	Gap (%)	CPU (s)	Avg. obj.	Std.	Gap (%)
10	0.519	330*	330	0	42.841	330	0	0
20	65.289	531*	531	0	59.887	535.4	1.768	0.83
30	7200	841	820.63	2.42	60.174	839.1	3.752	2.25
60	7200	1871	1690.13	9.67	62.339	1743.9	12.773	3.18
90	7200	3185	2460.26	22.75	66.907	2553.8	12.216	3.80
120	7200	4713	3334.27	29.25	70.781	3490.3	20.054	4.68
180	7200	–	–	–	85.130	5082.3	27.686	–

Avg. obj: average objective value; CPU: average CPU time obtained after 10 runs; Std.: standard deviation of objective values.
* Optimal value.

Table 4
Average running times with different combinations of penalty coefficients.

Unit penalty				CPU time (s)
up_i^1 $(\forall i)$	up_i^2 $(\forall i)$	$vp_i^{1,2}(\forall i)$	$op^{1,2}$	
1	1	1	1	68.082
10	1	1	1	68.473
1	10	1	1	66.317
1	1	10	1	67.598
1	1	1	10	69.520
100	1	1	1	70.910
1	100	1	1	69.551
1	1	100	1	69.035
1	1	1	100	68.754

as in Cherkesly et al. (2015). For the sake of computation efficiency, we set $IT_{edu} = 20$. The termination criterion is set to $(IT_{NI}, T_{max}) = (5000,\ 30\ min)$ as in Vidal et al. (2013).

4.4.2. Comparison of the performance of the exact method and the combined hybrid GA

Table 3 shows the running times (CPU) in seconds, the upper bounds (UBs), the lower bounds (LBs) and the gaps obtained by CPLEX; CPLEX stopped after reaching a 2 h limit or when an optimal solution was found. Table 3 also shows the average computation time (s) and the average and standard deviation of the objective values obtained by the combined hybrid GA in 10 runs, as well as the gap (%) representing the deviation of the average objective value from the LB.

When $|N| = 10$, CPLEX obtained an optimal solution in less than 1 s, whereas the combined hybrid GA took more than 40 s to obtain the same solution. The latter took more time because it could not stop immediately even when an optimal solution was found, but instead could stop only after a predetermined number of iterations (5000 in this case) without further improvement. This is a common problem with heuristics.

When $|N| = 20$, CPLEX still obtained an optimal solution in slightly over 1 min, whereas the combined hybrid GA could obtain a good, feasible solution with a gap of 0.83% in slightly less than 1 min.

For the larger problems ($|N| = 30$, 60, 90, 120), CPLEX only obtained a feasible solution and a lower bound in 2 h, and the gap of CPLEX enlarged when $|N|$ increased. The gap was greater than 20% when $|N|$ was at least 90. Meanwhile, the combined hybrid GA obtained a more feasible solution in only about 60 to 70 s. The gap obtained by the combined hybrid GA increased slightly with the problem sizes, with a gap of less than 5% for $|N| = 120$.

When $|N| = 180$, the combined hybrid GA obtained a feasible solution in about 85 s, whereas the exact method was unable to do so in 2 h. This shows the limitations of the exact method and the strength of our proposed method in large applications.

To sum up, the combined hybrid GA yields better solutions in shorter running times when $|N| \geq 30$. Overall, this method produces high-quality solutions in short computing times.

4.5. Effect on different penalties on the computation speed

To test the influence of different unit penalties on the computation speed of the proposed method, we consider a medium network with 90 stations. The parameter setting is the same as that in Section 4.4. We set the base scenario for comparison with all unit penalties set at 1. In the other cases, one of the unit penalties is set at 10 or 100. The average computation time of each case in 10 runs was computed and is given in Table 4. The results show that an increase in the magnitude for each unit penalty coefficient does not make a significant change in computation time.

5. Conclusions

We examine the SBRP with multiple types of bikes and with the use of substitution and occupancy strategies. We formulate the problem as a mixed-integer programming problem and develop a combined hybrid GA method to solve the problem. In this method, the HGSADC is modified to generate routing sequences, and a proposed greedy method is used to determine the loading/unloading instructions at each visited station, the substitution and occupancy strategies, and the vehicle load along each arc on each route. The results show that the combined hybrid GA yields high-quality solutions with short running times and that the unit penalties have a negligible effect on the computation time with this method. We also use small examples to illustrate the following. First, depending on the parameter settings, a substitution strategy may be adopted across stations or within a station, and a waiting strategy for the occupancy strategy may be applied. Second, optimal solutions can be sensitive to the unit penalties for imbalance and substitution. Third, the substitution and occupancy strategies may not be used simultaneously, even when both are allowed. When either strategy is allowed, the total cost may be reduced.

Further studies should include extension of the current problem and solution method to consider multiple vehicle cases, multiple visits, and dynamic repositioning problems. Moreover, it is important to propose accuracy methods to determine the unit penalties associated with substitution and the occupancy strategies, because these penalties may greatly affect the operational decisions. This is also left for future studies.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China (71271183, 71431003, 71571150, 71361006), the Humanities and Social Science Foundation of The Ministry of Education (14YJA630026), the Fundamental Research Funds for the Central Universities (26815WCX03), and the Key Research Base of Philosophy and Social Sciences in Sichuan Province (QGXH15-05). The authors are grateful to the reviewer for his/her constructive comments.

References

Alvarez-Valdes, R., Belenguer, J.M., Benavent, E., Bermudez, J.D., Muñoz, F., Vercher, E., Verdejo, F., 2015. Optimizing the level of service quality of a bike-sharing system. Omega doi:10.1016/j.omega.2015.09.007.

Angeloudis, P., Hu, J., Bell, M.G.H., 2014. A strategic repositioning algorithm for bicycle-sharing schemes. Transportmetrica A 10 (8), 759–774.

Anily, S., Hassin, R., 1992. The swapping problem. Networks 22 (4), 419–433.

Benchimol, M., Benchimol, P., Chappert, B., de la Taille, A., Laroche, F., Meunier, F., Robinet, L., 2011. Balancing the stations of a self-service bike hire system. RAIRO-Oper. Res. 45 (1), 37–61.

Brinkmann, J., Ulmer M.W., Mattfeld D.C., 2015. Inventory routing for bike sharing systems. https://www.tu-braunschweig.de/Medien-DB/winfo/publications/wp_brinkmann_inventory_routing_bike_sharing.pdf.

Caggiani, L., Ottomanelli, M., 2013. A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems. Procedia Soc. Behav. Sci. 87, 203–210.

Chajakis, E.D., Guignard, M., 2003. Scheduling deliveries in vehicles with multiple compartments. J. Global Optim. 26 (1), 43–78.

Chalasani, P., Motwani, R., 1999. Approximating capacitated routing and delivery problems. SIAM J. Comput. 28 (6), 2133–2149.

Chemla, D., Meunier, F., Calvo, R.W., 2013. Bike sharing systems: solving the static rebalancing problem. Discrete Optim. 10 (2), 120–146.

Chemla, D., Meunier, F., Pradeau, T., Calvo, R.W., Yahiaoui, H., 2013b. Self-service bike sharing systems: simulation, repositioning, pricing. Working paper.

Cherkesly, M., Desaulniers, G., Laporte, G., 2015. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. Comput. Oper. Res. 62, 23–35.

Christiansen, M., Fagerholt, K., Flatberg, T., Haugen, Ø., Kloster, O., Lund, E.H., 2011. Maritime inventory routing with multiple products: a case study from the cement industry. Eur. J. Oper. Res. 208 (1), 86–94.

Contardo, C., Morency, C., Rousseau, L.M., 2012. CIRRELT, Technical Report CIRRELT-2012-09. CIRRELT, Montreal, Canada.

Cornillier, F., Boctor, F., Renaud, J., 2012. Heuristics for the multi-depot petrol station replenishment problem with time windows. Eur. J. Oper. Res. 220 (2), 361–369.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S., 2014. The bike sharing rebalancing problem: mathematical formulations and benchmark instances. Omega 45, 7–19.

Erdoğan, G., Cordeau, J.F., Laporte, G., 2010. A branch-and-cut algorithm for solving the non-preemptive capacitated swapping problem. Discrete Appl. Math. 158 (15), 1599–1614.

Erdoğan, G., Laporte, G., Calvo, R.W., 2012. The one commodity pickup and delivery traveling salesman problem with demand intervals. Working Paper.

Erdoğan, G., Laporte, G., Calvo, R.W., 2014. The static bicycle relocation problem with demand intervals. Eur. J. Oper. Res. 238 (2), 451–457.

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

Hernández-Pérez, H., Rodrıguez-Martına, I., Salazar-Gonzáleza, J.J., 2015. A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem. Eur. J. Oper. Res. Available online.

Hernández-Pérez, H., Salazar-González, J.J., 2009. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. Eur. J. Oper. Res. 196 (3), 987–995.

Ho, S.C., Szeto, W.Y., 2014. Solving a static repositioning problem in bike-sharing systems using iterated tabu search. Transport. Res. E 69, 180–198.

Ho, S.C., Szeto, W.Y., 2016. GRASP with path relinking for the selective pickup and delivery problem. Expert Syst. Appl. 51 (1), 14–25.

Holland, J.H., 1975. Adaptation in Nature and Artificial System. MIT Press, Cambridge, MA.

Kloimüllner, C., Papazek, P., Hu, B., Raidl, G.R., 2014. Balancing bicycle sharing systems: an approach for the dynamic case. In: Evolutionary Computation in Combinatorial Optimisation. Springer, Berlin Heidelberg, pp. 73–84.

Lahyani, R., Coelho, L.C., Khemakhem, M., Laporte, G., Semet, F., 2015. A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. Omega 51, 1–10.

Lin, J.H., Chou, T.C., 2012. A geo-aware and VRP-based public bicycle redistribution system. Int. J. Veh. Technol. Article ID 963427.

Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. Transport. Res. B 89, 19–42.

Muyldermans, L., Pang, G., 2010. On the benefits of co-collection: experiments with a multi-compartment vehicle routing algorithm. Eur. J. Oper. Res. 206 (1), 93–103.

Nair, R., Miller-Hooks, E., Hampshire, R.C., Bušić, A., 2013. Large-scale vehicle sharing systems: analysis of Vélib'. Int. J. Sustain. Transport. 7 (1), 85–106.

Papazek, P., Raidl, G.R., Rainer-Harbach, M., Hu, B., 2013. A PILOT/VND/GRASP hybrid for the static balancing of public bicycle sharing systems. In: Computer Aided Systems Theory-EUROCAST 2013. Springer, Berlin Heidelberg, pp. 372–379.
Papazek, P., Kloimüllner, C., Hu, B., Raidl, G.R., 2014. Balancing bicycle sharing systems: an analysis of path relinking and recombination within a GRASP hybrid. In: Parallel Problem Solving from Nature PPSN XIII, Lecture Notes in Computer Science, 8672. Springer, pp. 792–801.
Pfrommer, J., Warrington, J., Schildbach, G., Morari, M., 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Trans. Intell. Transp. Syst. 15 (4), 1567–1578.
Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. 31 (12), 1985–2002.
Psaraftis, H.N., 2011. A multi-commodity, capacitated pickup and delivery problem: the single and two-vehicle cases. Eur. J. Oper. Res. 215 (3), 572–580.
Rainer-Harbach, M., Papazek, P., Hu, B., Raidl, G.R., 2013. Balancing bicycle sharing systems: a variable neighborhood search approach. In: Evolutionary Computation in Combinatorial Optimization. Springer, Berlin Heidelberg, pp. 121–132.
Raviv, T., Tzur, M., Forma, I.A., 2013. Static repositioning in a bike-sharing system: models and solution approaches. EURO J. Transport. Logist. 2 (3), 187–229.
Reed, M., Yiannakou, A., Evering, R., 2014. An ant colony algorithm for the multi-compartment vehicle routing problem. Appl. Soft Comput. 15, 169–176.
Relvas, S., Magatão, S.N.B., Barbosa-Póvoa, A.P.F., Neves, F., 2013. Integrated scheduling and inventory management of an oil products distribution system. Omega 41 (6), 955–968.
Rodríguez-Martín, I., Salazar-González, J.J., 2011. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem: a matheuristic. In: Network Optimization. Springer, Berlin Heidelberg, pp. 401–405.
Salazar-González, J.J., Santos-Hernández, B., 2015. The split-demand one-commodity pickup-and-delivery travelling salesman problem. Transport. Res. B 75, 58–73.
Schuijbroek, J., Hampshire, R., van Hoeve, W.J., 2013. Inventory rebalancing and vehicle routing in bike sharing systems. Tepper School of Business, Paper 1491, http://repository.cmu.edu/tepper/1491 (2016-02-12).
Szeto, W.Y., Liu, Y., Ho, S.C., 2016. Chemical reaction optimization for solving a static bike repositioning problem. Transport. Res. D doi:10.1016/j.trd.2016.05.005.
Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. Oper. Res. 60 (3), 611–624.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Comput. Oper. Res. 40 (1), 475–489.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2014. A unified solution framework for multi-attribute vehicle routing problems. Eur. J. Oper. Res. 234 (3), 658–673.