Dynamic green bike repositioning problem – A hybrid rolling horizon artificial bee colony algorithm approach

C.S. Shui and W.Y. Szeto

Department of Civil Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
The University of Hong Kong Shenzhen Institute of Research and Innovation, Shenzhen, China

Abstract

This paper introduces a new dynamic green bike repositioning problem (DGBRP) that simultaneously minimizes the total unmet demand of the bike-sharing system and the fuel and CO_2 emission cost of the repositioning vehicle over an operation period. The problem determines the route and the number of bikes loaded and unloaded at each visited node over a multi-period operation horizon during which the cycling demand at each node varies from time to time. To handle the dynamic nature of the problem, this study adopts a rolling horizon approach to break down the proposed problem into a set of stages, in which a static bike repositioning sub-problem is solved in each stage. An enhanced artificial bee colony (EABC) algorithm and a route truncation heuristic are jointly used to optimize the route design in each stage, and the loading and unloading heuristic is used to tackle the loading and unloading sub-problem along the route in a given stage. Numerical results show that the EABC algorithm outperforms Genetic Algorithm in solving the routing sub-problem. Computation experiments are performed to illustrate the effect of the stage duration on the two objective values, and the results show that longer stage duration increases total unmet demand and the total fuel and CO_2 emission cost. Numerical studies are also performed to illustrate the effects of the weight and the loading and unloading times on the two objective values and the tradeoff between the two objectives.

Keywords: green bike repositioning problem, dynamic bike repositioning problem, rolling horizon approach, artificial bee colony algorithm, vehicle emissions

1. Introduction

Bike-sharing systems (BSSs) are evolving worldwide. They provide numerous advantages such as reducing the short-distance motorized trips, complementing public transport, and reducing the greenhouse gas emissions. These systems escalate bikes to become a convenient and efficient transport mode by offering an automatic rental use of bikes in all bike-sharing stations within a city and allowing the users to return the bikes in any stations. Due to the characteristics of the stations, such as altitude, proximity to the public transport stations, time of day, or the regions they sited, some stations have bike surpluses while some stations have bike deficits. With bike deficiency, some cyclists cannot rent bikes at those deficit stations, leading to unmet demand. Therefore, the BSS operators need to redistribute their bikes among stations regularly to minimize the demand dissatisfaction. This redistribution can be done by employing vehicles to pick up their bikes from bike surplus stations to bike deficient stations. The redistribution problem is currently known as a bike repositioning problem (BRP).

The aim of a BRP is to determine optimal truck routes and the loading/unloading activities of each truck at stations based on the design objective, subject to various constraints related to the repositioning vehicles, stations, and operational constraints. This problem is more complicated than the classical vehicle routing problem (VRP) and the classical traveling salesman problem (TSP) because the repositioning problem further requires determining the pick-up and drop-off quantities at each station (Ho & Szeto, 2014).

The unique problem setting of a BRP has attracted the interest of many researchers in recent years. Table 1 has summarized the BRP publications according to their operation types and design objectives. In terms of operation type, the problems can broadly be classified into two classes: static and dynamic. The static problem considers nighttime operations in which station demand variations are negligible, while the dynamic problem considers daytime operations and real-time station demand variations. As seen in Table 1, a large portion of studies focuses on static BRPs while very few studies have addressed on dynamic BRPs. The contrast in the number of publications is mainly due to the difficulty in handling the varying demand during the operation period. In dynamic BRPs, the routes need to be updated regularly to resolve the demand variations arisen from time to time. Table 1 also illustrates that the existing studies adopt various objectives, such as minimizing vehicle travel time or cost (e.g., Benchimol et al., 2011; Chemla et al., 2012; Lin & Chou, 2012), minimizing total unmet demand (e.g., Contardo et al., 2012; Szeto et al., 2016), minimizing maximum tour length (Schuijbroek et al., 2017), minimizing the sum of travel and handling costs (Erdoğan et al., 2014), minimizing total relocation and lost user cost (e.g., Caggiani & Ottomanelli, 2012), and minimizing a weighted sum of total travel time, the total absolute deviation from perfect balance at each station, and the total number of loading and unloading quantities (e.g., Raidl et al., 2013; Rainer-Harbach et al., 2013, 2015). From these reviewed papers, the objective highlighted most is to minimize the total absolute deviation from perfect balance in public bike sharing systems, either by directly determining the number of bikes or indirectly in the form of penalty functions. Minimizing total unmet demand is a similar objective while it only focuses on bike deficits and neglects bike surpluses. These studies show that total unmet demand is a crucial indicator for the repositioning activity, but there are other important considerations for an optimal repositioning strategy. Specifically, Wiersma (2010) highlighted the threat of bike repositioning activities by vehicles to the environmental creditability of bike sharing systems, given that the bikes are generally relocated by fossil-fueled vehicles. Therefore, a repositioning plan that solely focuses on minimizing total unmet demand may result in long repositioning routes or heavy vehicle loads, which may adversely affect the environment by producing more air pollutants. However, to the best of our knowledge, no existing BRP studies have considered environmental needs (or green elements) in their design objectives.

Table 1 Summary of the characteristics of BRPs in the existing studies

Reference	**Type**	**Objectives**
Benchimol et al. (2011)	Static	Minimize total travel time
Caggiani & Ottomanelli (2012)	Dynamic	Minimize relocation and lost user cost
Chemla et al. (2012)	Static	Minimize total travel cost
Contardo et al. (2012)	Dynamic	Minimize total unmet demand
Lin & Chou (2012)	Static	Minimize total transportation cost (time or distance)
Chemla et al. (2013)	Static	Minimize total travel distance
Di Gaspero et al. (2013)	Static	Minimize the weighted sum of total travel time and total absolute deviation from the target number of bikes
Nair et al. (2013)	Static	Minimize total redistribution cost
Raidl et al. (2013)	Static	Minimize the weighted sum of total absolute deviation from the target number of bikes, total number of

		loading/unloading activities, and overall time required for all routes
Rainer-Harbach et al. (2013)	Static	Minimize the weighted sum of total absolute deviation from the target number of bikes, total number of loading/unloading activities, and overall time required for all routes
Raviv et al. (2013)	Static	Minimize the weighted sum of total travel time and penalty cost
Schuijbroek et al. (2017)	Static	Minimize maximum tour length
Erdoğan et al. (2014)	Static	Minimize travel and handling costs
Ho & Szeto (2014)	Static	Total penalty cost
Kloimüllner et al. (2014)	Dynamic	Minimize firstly unfulfilled demand and absolute deviation from the target fill level, and then total number of loading instructions and total drive time
Forma et al. (2015)	Static	Minimize the weighted sum of the expected number of unserved users during the next working day and total travel distance
Rainer-Harbach et al. (2015)	Static	Minimize the weighted sum of total absolute deviation from the target number of bikes, total number of loading/unloading activities, and overall time required for all routes
Li et al. (2016)	Static	Minimize the sum of total vehicle travel cost, total unbalanced penalty costs for all bike types, total substitution penalty cost, and total occupancy penalty cost
Szeto et al. (2016)	Static	Minimize the weighted sum of unmet customer demand and operational time
Ho & Szeto (2017)	Static	Minimize the weighted sum of total travel time and penalty cost
This study	Dynamic	Minimize the weighted sum of total unmet demand and total fuel and CO_2 emission cost

As a first step to consider environmental objectives, this study proposes a new problem, referred to as a dynamic green bike repositioning problem (DGBRP). This problem considers total carbon dioxide (CO_2) emission related cost as the environmental objective, which has been widely adopted in other types of green logistic problems (e.g., Demir et al., 2012; Koç et al., 2014), as CO_2 is regarded as one of the most serious threats to the environment through the greenhouse effect (Ericsson et al., 2006) and road transport, especially road freight transport, which constitutes a large portion of CO_2 emissions (Jabali et al., 2012). As bike repositioning is also a logistical activity that heavily relies on fossil-fueled vehicles (Wiersma, 2010), minimizing total CO_2 emission related cost is a representative and significant environmental objective for the DGBRP.

With respect to the emission minimization objective, the DGBRP is similar to a pollution routing problem (PRP), in which the designated route should minimize the pollutant emissions. For a PRP, its design objective is to minimize emission cost, plus other costs if any; and the quantities of commodities delivered to each customer (or node) are known at the beginning of the operation. In other words, some routing problems that are not named as PRPs should be considered to be PRPs or their variants, given that they consider emission minimization (or fuel consumption minimization or the corresponding cost minimization) to be the sole objective or one of the objectives in their design problems, and the quantities delivered to each node are known at the beginning of the operation. Examples include eco-routing problems (e.g., Ericsson et al., 2006) and emission vehicle routing problems (e.g., Figliozzi, 2010; Kopfer et al., 2014). The proposed DGBRP, however, differs from the abovementioned problems in several ways. First, the pickup or drop-off locations are not given, and therefore any node (i.e., any station and the depot) can be a source or destination of bikes. Second, the pickup or drop-off quantity at each node is a decision variable and has an effect on the objective function value (i.e., total unmet demand and the total fuel and CO_2 emission cost). Third, the pickup or drop-off quantity at each node varies with respect to time. These three points distinguish the DGBRP from existing PRPs.

To measure the CO_2 emissions or their cost, the conventional approach is to determine the vehicle emissions based on existing fuel consumption models as the emissions are directly proportional to the fuel consumption (Demir et al., 2012). As greenhouse gas emissions from transportation have received attention for a long while, there has been a wide range of fuel consumption and vehicle emission models in the literature. Demir et al. (2014) categorized fuel consumption (vehicle emission) models into three main groups with respect to data complexity: factor models, macroscopic models, and microscopic models. Factor models adopt simple fuel consumption methods, e.g., the distance-based method introduced in GHG Protocol (2013) and the emission factor calculation by DEFRA (2012), to convert fuel consumption to vehicle emissions, and are particularly useful when the information of traffic flow and operation is insufficient. Due to the lower level of data complexity, factor models are adopted in some green logistic problems (e.g., Kopfer et al., 2014; Zhang et al., 2014). In macroscopic models, the fuel consumption is generally formulated as a function of average speed, which reflects the empirical findings that the fuel consumption rate varies with respect to speed (e.g., Demir et al., 2011) and is consistent with the vehicle emission literature (Szeto et al., 2012). Examples for popular macroscopic models include MEET[1], COPERT[2], MOVES[3], and HBEFA[4]. In particular, COPERT is often used in some emission reduction projects, such as AMITRAN (AMITRAN, 2016), PRIMES (and also PRIMES-TREMOVE) (E3M Lab, 2014), EMEP/EEA air pollutant emission inventory guidebook 2016 (European Environment Authority, 2016), and ICT-emission (Vock et al., 2014). Microscopic models include more instantaneous kinematic or aggregated modal variables than the macroscopic models to predict fuel consumption and vehicle emissions more accurately. Readers can refer to the review by Demir et al. (2014) which listed 12 existing microscopic emission (fuel consumption) models and the attributes associated with the models. Among all microscopic models, CMEM[5] (presented by Barth et al. (2005)) has been the most popular one used to determine fuel consumption or vehicle emissions in green logistic problems (e.g., Bektaş & Laporte, 2011; Jabali et al., 2012). Furthermore, as highlighted by Demir et al. (2014), simple factor models, MEET, COPERT, and CMEM are commonly adopted emission models in the existing green logistic studies.

In this DGBRP, we use the modified version of CMEM proposed by Demir et al. (2012) to determine the fuel consumption and hence the total fuel and CO_2 emission cost associated with the vehicle instead of other models. There are five reasons behind our choice of this model: (1) it is a function of the payload of the vehicle. This relationship is not found in many of the existing models, such as MEET and COPERT; (2) it considers the link-specific vehicle speed, another significant yet unique component for each road link which directly contributes to emissions; (3) it includes vehicle travel distance for each link, an operational attribute that is recognized to influence vehicle emissions (GHG Protocol, 2013); (4) compared with a

[1] Methodology for Calculating Transport Emissions and Energy Consumption (Hickman et al., 1999)

[2] Computer Programme to calculate Emissions from Road Transport (Ntziachristos et al., 2009)

[3] Motor Vehicle Emission Simulator (United States Environmental Protection Agency, 2010)

[4] Handbook Emission Factors for Road Transport (Hausberger et al., 2009)

[5] Comprehensive Modal Emission Model (Barth et al., 2005)

similar model introduced by Bektaş & Laporte (2011), it can determine the vehicle emissions at low speeds (i.e., speed smaller than 40 km/h). Given that the average vehicle speed in an urban region during daytime is always lower than 40 km/h for trucks, the chosen model can determine the emissions at those speeds; (5) it is one of the commonly used models in the existing green logistic studies.

In this modified model, the fuel consumption is the sum of two components: the fuel consumption associated with the engine performance and the tractive power. The fuel consumption associated with engine performance is inversely proportional to vehicle speed, while the fuel consumption associated with the tractive power is jointly proportional to the vehicle payload and inversely proportional to speed. When the average vehicle speed on each link is known, the fuel consumption is directly proportional to the vehicle payload. According to this relationship, holding a large number of bikes on a repositioning vehicle or visiting more nodes may not be an optimal strategy from the perspective of vehicle emissions. Nevertheless, holding too few bikes on vehicles or visiting too few nodes may not reduce the demand dissatisfaction significantly. This study, therefore, investigates the trade-off between demand dissatisfaction and emission (cost) minimization objectives in this DGBRP.

This paper focuses on a dynamic BRP, which considers a static BRP as a sub-problem, which in turn is an extension of the selective pickup and delivery problem. As highlighted by Ting & Liao (2013) and Ho and Szeto (2016), the selective pickup and delivery problem is already NP-hard. Moreover, Benchimol et al. (2011) also proved that their static BRP for general networks and inputs is also NP-hard. Therefore, our proposed problem with a refined objective function, which also captures the selective pickup and delivery problem as a sub-problem and has a similar static repositioning sub-problem, is also NP-hard. Also, the problem in this paper is new and non-linear with respect to the introduction of the environmental objective. The non-linear problem nature disables the *direct* use of the existing exact methods shown in Table 2, which are often used for solving *linear* integer programming problems. Moreover, it is inefficient to use exact methods such as those shown in Table 2 to solve large, realistic repositioning problems in general. Furthermore, the existing heuristics or approximation methods shown in Table 2 are tailored for solving their bike repositioning problems that are different from ours in terms of constraints. For example, our problem has a new set of constraints that allow the vehicle to travel to more than one node in each period and require the vehicle to travel until passing the beginning of the next period. These constraints allow the travel times between arcs not to be necessarily equal to the multiples of time periods. These constraints significantly increase the problem complexity. Therefore, their methods cannot be directly applied to solve our problem and new methods are needed to develop to cater for these constraints.

As our ultimate objective is to develop a solution method that can solve large and realistic bike repositioning problems efficiently, we prefer to develop a heuristic to solve the proposed problem. As seen from Table 2, only classical metaheuristics, including ant colony optimization and tabu search, are adopted in solving BRPs, while some recent metaheuristics have not been considered in these studies. As a recently developed metaheuristic, the enhanced artificial bee colony (EABC) algorithm proposed by Szeto et al. (2011) is used in this paper to determine a route for the repositioning vehicle. The EABC algorithm is not bounded by the mathematical properties of the objectives so it can find nearly optimal solutions with much shorter computational time compared with other existing heuristics when proper methods to handle solution feasibility are introduced. The EABC algorithm and its former version, the artificial bee colony (ABC) algorithm, have the advantages of easy application and prove their capability of handling various discrete and combinatorial problems, such as leaf-constrained minimum spanning tree problems (e.g., Singh, 2009), knapsack problems (e.g., Sundar et al. 2010), TSPs (e.g., Karaboga & Gorkemli, 2011), periodic VRPs (e.g., Yao et al., 2013), capacitated VRPs (e.g., Szeto et al., 2011; Alvarado-Iniesta et al., 2013), and job shop scheduling problems (e.g., Li et al., 2011; Pan et al., 2011). Providing that the potential of ABC and EABC algorithms are still under exploration, it is anticipated that they are efficient solution methods for other problems. Therefore, the EABC algorithm is used to generate the vehicle route, but it can be replaced by other routing heuristics without conceptual difficulty.

To solve the DGBRP, this paper adopts a rolling horizon approach to handle the time-varying demand. This approach is useful for real-time traffic assignment (Peeta & Mahmassani, 1995), tour scheduling problems (Stolletz & Zamorano, 2014), and vehicle routing problems for auto-carriers (Cordeau et al., 2015). The main idea is to decompose the repositioning route for the whole service time horizon into smaller but well-connected sub-problems that cover only part of the entire service horizon. By this approach, the dynamic

BRP can be decomposed into a set of static BRPs for each fixed duration (consisting of a number of periods), with the demand and inventory levels updated after each period, which can be solved more easily. To the best of our knowledge, no previous dynamic BRP study has adopted a rolling planning horizon approach to solve the proposed problem at the time of this writing. To deal with the fixed duration for each static problem, we propose a route truncation heuristic for revising the route generated within the EABC algorithm in which the route duration is longer than the fixed duration. This heuristic invokes a loading and unloading heuristic from time to time to determine the loading and unloading quantities for determining the service time required at each node on the concerned route during the calculation of the route duration.

Table 2 Summary of the solution methods for BRPs

Approach	**Solution Method**	**Reference**
Exact method	Branch-and-cut algorithm	Erdoğan et al. (2014); Dell'Amico et al. (2014)
Approximation	9.5-approximation algorithm	Benchimol et al. (2011)
Heuristics/metaheuristics	Cluster-first route-second	Schuijbroek et al. (2017)
	Ant colony + constraint programming	Di Gaspero et al. (2013)
	Iterated tabu search	Ho & Szeto (2014)
	PILOT+ variable neighbor descent, GRASP + variable neighbor descent	Kloimüllner et al. (2014); Rainer-Harbach et al. (2014)
	Variable Neighborhood Search (VNS)	Raidl et al. (2013); Rainer-Harbach et al. (2014); Kloimüllner et al. (2014)
	Chemical reaction optimization	Szeto et al. (2016)
	Destroy and repair algorithm	Dell' Amico et al. (2016)
	Hybrid genetic algorithm	Li et al. (2016)
	Hybrid large neighborhood search	Ho & Szeto (2017)
Hybrid exact and heuristics	Branch-and-cut algorithm with tabu search	Chemla et al. (2013)
	Cluster-first, cluster-route-second, route-third heuristic	Forma et al. (2015)

To summarize, the main contributions of this paper are the following:

1. We propose a solution framework for solving the DGBRP. The dynamic problem is

decomposed into several static BRP sub-problems by the rolling horizon approach. An EABC algorithm and a proposed route truncation heuristic are jointly adopted to determine vehicle route in each static sub-problem, and a loading and unloading heuristic is used to solve the loading and unloading sub-problem along a given route; and

2. we investigate the trade-off between total unmet demand and the total fuel and CO_2 emission cost and illustrate the properties of the problem through sensitivity analysis.

The remainder of this paper is organized as follows. Section 2 discusses the problem setting and the fuel consumption and CO_2 emission cost model of the DGBRP. Section 3 describes the solution framework to solve the DGBRP. Section 4 illustrates the performance of the proposed solution method and the effect of varying problem parameters on total unmet demand and the total fuel and CO_2 emission cost. Finally, Section 5 gives conclusions.

2. Problem description

2.1. Problem setting

We consider a network N that has one depot (denoted by 0) only and multiple bike stations. The whole study horizon $\left[0,H\right]$ is discretized into $|T|$ equal periods of length l such that $H=l\left|T\right|$, where T is a set of periods and $|T|\geq 2$. A vehicle with its capacity Q is employed to reposition bikes among the stations and the depot within the study horizon. The rebalancing strategy is determined and updated at the beginning of each period t based on the latest information or prediction of the demand and the number of bikes available (i.e., inventory level) at each node for the current period t and next n_t periods, where $0\leq n_t\leq |T|-t$. As the demand and inventory levels at the nodes (including stations and the depot) are changing across periods, the vehicle is allowed to visit a node multiple times to perform the rebalancing operation. The vehicle starts from the depot, travels to the assigned nodes to perform loading and unloading, and finally returns to the depot at the end of the operation. The vehicle is also allowed to visit the depot to load and unload bikes throughout the whole study horizon and the depot is assumed to always have enough bikes for pickup and docks for bike storage. It is also assumed that each period is long enough so that the vehicle can visit at least one node in a period. Because the demand is assumed to remain unchanged during each period, the vehicle only requires visiting each node at most once in each period. The vehicle is not allowed to wait at any node except for loading and unloading. However, the vehicle can wait at the depot in the last period $|T|$ even after the unloading process is finished because the remaining operational time may not be enough to serve other stations. The loading and unloading times for each bike at each node are fixed. The distance of each arc and the average vehicle speed on each arc in each period are given, and thus the travel time of each arc in each period can be deduced.

The problem aims to determine a route for the operating vehicle over the horizon and the loading and unloading quantities at each node during each period in order to minimize the weighted sum of penalty cost for the total unmet demand of the system and the fuel and CO_2 emission cost of the operating vehicle of each design interval, where a design interval consists of a period t in the horizon plus the following n_t periods.

2.2. Fuel consumption and CO_2 emission cost

To capture the environmental influence of the repositioning activity, we adopt the modified CMEM adopted by Demir et al. (2012), based on Barth et al. (2005), to estimate the instantaneous fuel consumption and then the corresponding fuel and CO_2 emission cost. According to that model, the fuel consumption rate R' (in gram/second) is given by

$$R' = \xi\left(hZV + P/\chi_0\right)/\kappa_0 , \tag{1}$$

where ξ is the fuel-to-air mass ratio, h is the engine friction factor, Z is the engine speed, V is the engine displacement, and χ_0 and κ_0 are constants. P is the engine power output (in kilowatt) and can be calculated as

$$P = P_{\mathrm{T}}/\eta_{\mathrm{F}} + P_{\mathrm{A}} , \tag{2}$$

where η_{F} is the vehicle drive train efficiency, and P_{A} is the engine power demand for energy losses of the engine and operation of vehicle accessories, which is assumed to be zero for simplicity. P_{T} is the total tractive power requirements (in kilowatt) of the wheels:

$$P_{\mathrm{T}} = \left(J\Omega + Jg\sin\theta + 0.5C_{\mathrm{D}}\Delta Av^2 + JgC_{\mathrm{R}}\cos\theta\right)v/1000 , \tag{3}$$

where J is the total vehicle weight (in kilogram), v is the vehicle speed (meter/second), Ω is the acceleration (meter/second^2), θ is the road angle, g is the gravitational constant, Δ is the air density, A is the frontal surface area of the vehicle, and C_{D} and C_{R} are the coefficients of the aerodynamic drag and rolling resistance, respectively.

For link (i, j) that joins nodes i and j, let d_{ij} be the length of the link (meter), and v_{ij} be the average speed traveling on this link. If all variables in equations (1)-(2) except the vehicle speed v_{ij} remain constant on this link, the fuel consumption (in liter), $E_{ij}\left(v_{ij}\right)$, on this link can be formulated as a function of v_{ij} and expressed as

$$E_{ij}\left(v_{ij}\right) = hZV\kappa_1 d_{ij}/v_{ij} + P_{\mathrm{T}}\kappa_1\chi_1 d_{ij}/v_{ij} \ , \tag{4}$$

where $\kappa_1 = \xi/\kappa_0\psi$ and $\chi_1 = 1/1000\eta_{\mathrm{F}}\chi_0$ are constants, and ψ is the conversion factor of fuel from gram/second to liter/second. Besides, let $J = w + S$, where w is the curb weight (i.e., the weight of an empty vehicle) and S is the payload on the vehicle. Let $\phi_1 = \Omega + g\sin\theta + gC_{\mathrm{R}}\cos\theta$ be a vehicle-link specific constant, and $\phi_2 = 0.5C_{\mathrm{D}}\Delta A$ be a vehicle-specific constant. By omitting the link indices (i, j) on S and ϕ_1 for simpler presentation, $E_{ij}\left(v_{ij}\right)$ can be rewritten as

$$E_{ij}\left(v_{ij}\right) = \kappa_1\left(hZV + w\chi_1\phi_1 v_{ij} + S\chi_1\phi_1 v_{ij} + \phi_2\chi_1 v_{ij}{}^3\right)d_{ij}/v_{ij} \ . \tag{5}$$

The parameters used in this fuel consumption model and their corresponding values are listed in Table 3, following the values for light duty vehicles given in Koç et al. (2014) as light duty vehicles are often employed in daytime bike repositioning. As the fuel consumption rate is determined, the fuel and CO$_2$ emission cost can be determined because the emission cost is directly proportional to the fuel consumption.

As highlighted in Bektaş & Laporte (2011) and Demir et al. (2012), the term related to hZV (i.e., the engine term) in (5) is significant at a low vehicle speed (i.e., less than 40 km/h), while the remaining terms (i.e., the terms associated with tractive power) are significant for higher speed levels. In the context of DGBRP, as the average vehicle speed in the urban region is seldom greater than 40 km/h (or approximately 11 m/s), this engine term must be considered. Meanwhile, we assume each bike has a weight of 17 kilograms, which is equivalent to the weight of a second-generation bike of YouBike (DIT Taipei, 2013).

Meanwhile, the vehicle is assumed to have an extra dead load of 160 kilograms other than the curb weight for the bike-related accessories, such as bike racks, and thus the vehicle can load a maximum of 20 bikes on each vehicle due to the maximum load constraint.

Table 3 Parameters used in the fuel consumption and CO_2 emission cost model

Notation	Description	Value
w	Curb weight of the vehicle, excluding the dead load of bike-related accessories (kilogram)	3500
w'	Curb weight of the vehicle, including the dead load of bike-related accessories (kilogram)	3660
J	Maximum total weight of the vehicle (kilogram)	4000
φ	Weight of a bike (kilogram)	17
ξ	Fuel-to-air mass ratio	1
h	Engine friction factor (kilojoule/rev/liter)	0.25
Z	Engine speed (rev/second)	38.34
V	Engine displacement (liters)	4.5
g	Gravitational constant (meter/ second^2)	9.81
C_D	Coefficient of aerodynamic drag	0.6
C_R	Coefficient of rolling resistance	0.01
Δ	Air density (kilogram/ meter^3)	1.2041
A	Vehicle frontal surface area (meter^2)	7.0
η_F	Vehicle drive train efficiency	0.45
χ_0	Efficiency parameter for diesel engines	0.45
κ_0	Heating value of a typical diesel fuel (kilojoule/ gram)	44
ψ	Conversion factor (from gram/second to liter/second)	737
Ω	Acceleration (meter/ sq. second)	0
θ	Road angle	0
f_C	Fuel and CO_2 emission cost per liter (£)	1.4

Based on the above notations, the fuel consumption of the vehicle that loads q bikes can be expressed as

$$E_{ij}\left(v_{ij}\right) = \kappa_1\left(hZV + w'\chi_1\phi_1 v_{ij} + \varphi q\chi_1\phi_1 v_{ij} + \phi_2\chi_1 v_{ij}{}^3\right) d_{ij}/v_{ij} \, . \tag{6}$$

In this equation, the term w' is the curb weight of the vehicle plus the dead load of the bike-related accessories (e.g., bike racks to place the bikes on the vehicle), and the term φq is the bike load on the vehicle. The fuel and CO_2 emission costs of the vehicle associated with link (i, j), e_{ij}, can then be expressed as

$$e_{ij} = f_C\kappa_1 b_{ij}\left(hZV + w'\chi_1\phi_{1,ij}v_{ij} + \varphi q_{ij,t}^{k}\chi_1\phi_{1,ij}v_{ij} + \phi_2\chi_1 v_{ij}^3\right), \tag{7}$$

where $b_{ij} = d_{ij} / v_{ij}$ represents the travel time on link (i, j).

3. A hybrid rolling horizon Artificial Bee Colony Algorithm

This study introduces a hybrid rolling horizon artificial bee colony algorithm to solve this DGBRP. The proposed algorithm is formed by four main components: the rolling horizon algorithm, the EABC algorithm, a proposed route truncation heuristic for route determination in each stage of the rolling horizon algorithm, and the loading and unloading (LU) heuristic. The first one is the main algorithm, which invokes the EABC algorithm from time to time, which in turn invokes the proposed route truncation heuristic and then the LU heuristic upon request.

3.1. The rolling horizon algorithm

Figure 1 explains the rolling horizon approach by showing two consecutive stages. In general, the planning horizon for the vehicle to travel is subdivided into several stages (i.e., design intervals), each of which consists of a roll period and the overlapping portion with the next stage (except the last stage with no overlapping portion). Each stage τ consists of (n_t+1) periods and the length of each roll period is l time units, where t is the index of the roll period in stage τ and is numerically equal to τ. Therefore, the time span ρ of each stage τ is (n_t+1) l time units, which is a multiple of the length of roll period l. In this study, all stages have the same number of consecutive periods, denoted as ρ', except that for the last (ρ' -1) stages, the number of consecutive periods is the number of periods remaining until the end of the modeling horizon. At the beginning of each stage, the demand for bikes and the inventory level at each node at the beginning of the roll period and the corresponding forecasts at the beginning of the following periods (i.e., the periods in the overlapping portion) are known and used in the main algorithm. The repositioning problem over each stage (τ-1) is a *static* problem and solved by the EABC algorithm (which invokes the proposed route truncation heuristic and the LU heuristic) by using the available information from that stage, but implemented only in the roll period of stage (τ-1). The start of projection horizon is then rolled forward by l time units to obtain the next stage τ. The demand and inventory level at each node are updated in the next stage τ and used to determine a new routing, loading, and unloading strategy in the new roll period in stage τ. This procedure is repeated until the end of the modeling horizon. Note that in this algorithm, the stage length (or the projection horizon) is not fixed to ensure that the end of each stage cannot be later than the end of the modeling horizon.

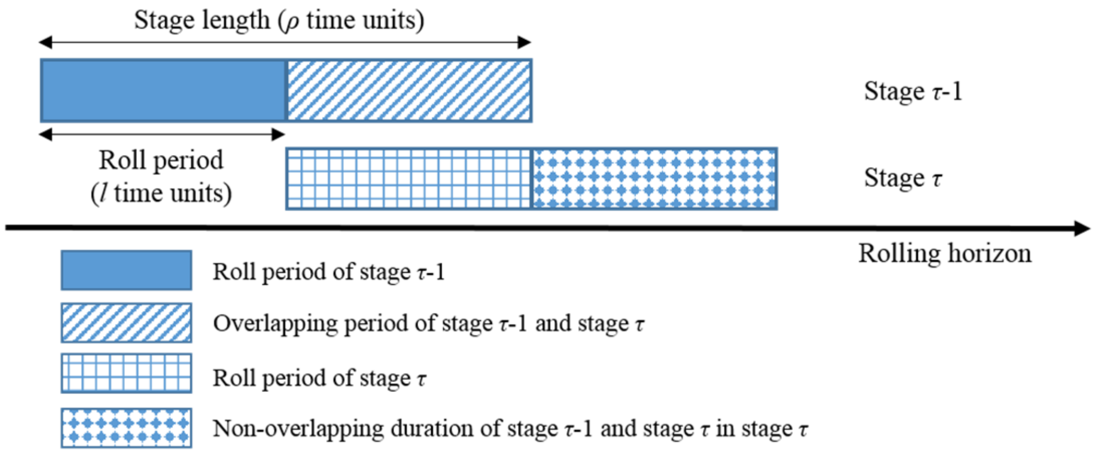
Stage length (ρ time units)
Stage τ-1
Roll period
(l time units)
Stage τ
Rolling horizon
Roll period of stage τ-1
Overlapping period of stage τ-1 and stage τ
Roll period of stage τ
Non-overlapping duration of stage τ-1 and stage τ in stage τ

Figure 1 Successive stages of the solution procedure

The outline of the solution procedure is provided as follows:

Step 1: Obtain the demand and inventory level at each node for each period in the first stage. Set $\tau = 1$.

Step 2: Determine a route (node sequence) in current stage τ by the EABC algorithm, a proposed route truncation heuristic, and the LU heuristic.

Step 3: Determine the loading and unloading activity at each visited node of the best route determined in Step 2 by the LU heuristic.

Step 4: Update the nodes' demand and inventory level in each period of stage τ. Store the route and the loading and unloading activities in stage τ that are not overlapped with the next stage. Update the starting position of the vehicle as the first node in the overlapping period and update the corresponding load on the vehicle at that node.

Step 5: If the current stage is not the last stage, then $\tau = \tau + 1$ and go to step 2. Otherwise, the algorithm stops and outputs the result.

3.2. The enhanced Artificial Bee Colony algorithm for the routing sub-problem

The static BRP in each stage is divided into a routing sub-problem and a loading and unloading sub-problem. The routing sub-problem determines the route of the repositioning vehicle in a given stage which consists of a sequence of visited nodes. In each stage, the vehicle visits at least one node. The route travel time is not necessarily equal to the multiples of time periods, and the lengths of all stages are not necessarily equal. The requirements for the starting and/ or ending positions of the route in different stages are listed below:

1. In the first stage, the route must start at the depot;
2. Except the first stage, the starting position of the route is the last visiting node of the route in the last stage;
3. In the last stage, the route must end at the depot.

As the above constraints have not been considered and handled in the literature, this paper develops a new method based on the enhanced Artificial Bee Colony algorithm to determine the route in each stage to cater for these constraints.

The enhanced Artificial Bee Colony (EABC) algorithm is an improved heuristic proposed by Szeto et al. (2011) based on the ABC algorithm, which is a swarm-based meta-heuristic algorithm introduced by Karaboga (2005). The ABC algorithm is developed based on the intelligent behavior of the honeybees' foraging process (Karaboga, 2009). Three types of bees, including employed bees, onlooker bees, and scout bees, are found in the ABC algorithm. Different types of bees play different roles in the exploration and exploitation of food sources. Food sources are considered to be solutions to specific problems or sub-problems.

This whole process of the ABC algorithm can be described as follows. In each iteration, each employed bee is firstly assigned to a food source. It collects the information of that food source (e.g., the nectar amount of the food source, equivalent to the fitness of the solution) and carries out a neighborhood search to find a better food source nearby. If a better source is found, the employed bee abandons the assigned food source and remembers the better one. After all of the employed bees have collected the information, they fly back to their hive to share the information of their assigned food sources with other unemployed onlooker bees. The onlooker bees choose to follow a certain employed bee based on a probabilistic selection and exploit the region near the corresponding food source (using a neighborhood operator) and calculate the nectar amount of the neighbor food source. Then, for each old food source, the best food source among all food sources near the old food source is determined. The employed bee associated with the old food source is assigned to the best food source and abandons the old food source if the best food source is better than the old food source. Otherwise, the employed bee still reports the old food source as its current best food source found. A food source is also abandoned by an employed bee if the quality of the food source has not been improved for *Limit* (a predetermined number) successive iterations. The employed bee abandons that food source and becomes a scout bee, which randomly finds a new food source to replace the old one. The actions of the three types of bees become the three major steps within an iteration in the ABC algorithm, namely (1) the employed bee phase, (2) the onlooker bee phase, and (3) the scout bee phase. A stopping criterion (generally the maximum number of iterations) is applied to terminate the foraging process.

The basic ABC algorithm can solve certain types of problems with great success, but there are rooms for improving its performance. Two improving mechanisms are introduced by Szeto et al. (2011) to enhance the performance of the ABC algorithm.

First, the current onlooker bee phase is to use a newly searched and better food source to replace the *corresponding* old neighbor food source. The improved mechanism is to replace a food source with a newly search food source by fulfilling two criteria: (1) the value of *Limit* of the food source being replaced is the largest among *all* existing food sources known, which implies that the replaced food source has not been improved for the largest number of times; and (2) the newly searched food source is better than the corresponding old neighbor food source and the food source identified by criterion (1). This modified approach gives more chance for potential food sources to be explored and excludes non-potential food sources which have not improved for a relatively large number of times and are worse than the new food sources (Szeto et al., 2011).

Second, in the scout bee phase of the basic ABC algorithm, the scout bees are sent to randomly search for a new food source to replace the old one that reaches the limit. Instead of random search, the modified procedure is to search for a new food source by applying a neighborhood operator to that old one, without evaluating its nectar amount. This approach can limit the search in bad food source regions with no control of the quality of food sources (Szeto et al., 2011).

Based on the above discussion, the procedure of the EABC algorithm used for solving the routing problem of the static BRP is summarized as follows:

Step 2.1. Food sources (i.e., routes) z_y are randomly generated, where $y = 1, \dots, Y$, and Y is the number of food sources.

Step 2.2. Each employed bee is assigned to a food source. Each food source's fitness value $f\left(z_y\right)$ is evaluated.

Step 2.3. Initialize $B = 0$ and $L_1 = L_2 = \ldots = L_Y = 0$,

where B = Number of times of repeating the whole foraging process;

L_y = Number of times of applying a neighborhood operator to food source y, y = 1, …, Y.

Step 2.4. The foraging process is repeated:

a. Employed Bee Phase

i. Each food source is applied by a neighborhood operator: $z_y \rightarrow \tilde{z}_y$.

ii. If $f\left(\tilde{z}_y\right) > f\left(z_y\right)$, z_y is replaced with $\tilde{z}_y$ and L_y = 0. Otherwise, L_y = L_y + 1.

b. Onlooker Bee Phase

i. Each onlooker bee selects a food source among all existing food sources using the roulette wheel selection method based on their fitness values.

ii. Each food source is applied by a neighborhood operator: $z_y \rightarrow \tilde{z}_y$.

iii. If $f(\tilde{z}_y) > f(z_y)$, select $z_{y'}$ to be replaced by $\tilde{z}_y$, where $L_{y'}$ is the maximum among all existing food sources and $f(\tilde{z}_y) > f\left(z_{y'}\right)$, and then set $L_{y'} = 0$. Otherwise, $L_y = L_y + 1$.

c. Scout Bee Phase

i. For each food source, if L_y = *Limit*, the food source is modified by a neighborhood operator: $z_y \rightarrow \tilde{z}_y$ and z_y is replaced by $\tilde{z}_y$.

ii. $B = B + 1$.

Step 2.5. Record the best route so far.

Step 2.6. The foraging process is stopped when B = Maximum number of iterations.

3.3. Route representation in the EABC algorithm

In the EABC algorithm, a route z_y is represented by a sequence of visited nodes (including stations and the depot). Each station is given an index number, e.g., 1 to 5, and the depot is represented by 0. For a single vehicle traveling across 5 nodes, one possible representation is $0 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 3$, which shows that the vehicle departs from the depot, passing through stations 1, 4, 5, 2, and 3 subsequently. For an intermediate stage starting at station 3, a possible route can be $3 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 6$. The route generated by the EABC algorithm and kept in the pool of food sources should be long enough for the vehicle to traverse during each stage considered. A long and fixed route length (measured by the number of nodes) is determined in advance, based on the comparison of average vehicle travel time between nodes (in minutes) and maximum stage length (in minutes). In the previous example, the long and fixed route length is 6.

3.4. Fitness evaluation and the proposed route truncation heuristic

The fitness of a route is directly related to the objective function. Because this function is also related to loading and unloading quantities at each node, these quantities must be determined before evaluating the fitness of a route. Moreover, the loading and unloading decisions significantly depend on and can be derived from the route. Furthermore, each route generated by the EABC algorithm is always longer than that required by a stage, and thus a method is required to determine the route in each stage. In addition, when determining the route in each stage, the actual loading and unloading times other than travel times between nodes should be taken into account to ensure that the stage length is not exceeded. Taking into account the preceding four considerations, we propose a route truncation heuristic to determine the route in each stage z'_y, loading and unloading quantities at each node, and the fitness of a route z_y.

For a route z'_y in a stage *τ*, we define *G* as the set of visited nodes on the route, i.e., $G \subset N$, and define $i_m \in G$, $m = 1,2,\ldots,|G|$, where the subscript *m* is used to define the order of nodes being visited in stage *τ*. Based on these notations, let p'_{i_m} and r'_{i_m} be the pickup and drop-off quantities at the m-th visited node, respectively, and $\overline{q}_{i_{m-1}i_m}$ and $\overline{e}_{i_{m-1}i_m}$ to be the number of bikes on the vehicle and the fuel and CO_2 emission cost, respectively, when the vehicle travels from the (m-1)-th to the m-th visited node of the route. Moreover, let Λ_j and Λ'_j represent the unmet demands at the visited and the non-visited nodes, respectively. Furthermore, let α be the weight for total unmet demand. Based on these notations, the procedure can be described as follows.

Step 2.2.1: Set $c_0 = 1, X = 1$.

Step 2.2.2: Generate a new route z'_y for a stage τ based on the first X nodes on the route concerned z_y (obtained from Step 2 of the EABC algorithm). If $\tau = |T|$, add the depot at the end of the new route.

Step 2.2.3: Solve the loading and unloading sub-problem by the LU heuristic (introduced in Section 3.5) based on z'_y to obtain the loading and unloading quantities at each node along the route of that stage, i.e., p'_{i_m} and r'_{i_m}.

Step 2.2.4: If the sum of the travel times and the loading and unloading times from the loading and unloading strategy (i.e., the operation time in the stage) from Step 2.2.3 does not exceed the "effective" stage length, set $c_0 = c_0 + 1$ and $X = X + 1$, and then go to Step 2.2.2.

Step 2.2.5: Calculate the number of bikes on the vehicle $\overline{q}_{i_{m-1}i_m}$ along the route.

Step 2.2.6: Evaluate the fitness of each route z_y by

$$f\left(z_y\right) = 1 \Big/ \left\{ \alpha \left(\sum_{j \in G} \Lambda_j + \sum_{j \in N \setminus G} \Lambda'_j \right) + \sum_{\gamma=1}^{X-1} \overline{e}_{i_\gamma i_{\gamma+1}} \right\}.$$

To reduce the computation time for Step 2.2.1, X can be initialized to a positive value based on the estimated sum of in-vehicle travel times and loading and unloading times of the first X nodes on the route z_y. In this estimation, for the loading and unloading times, this paper adopts a greedy method—the vehicle is assumed to 1) load as many bikes as possible at each bike surplus node (i.e., a station with more bikes available to serve the cycling demand there) onto the vehicle until the vehicle is full or all bikes at that node are loaded, and 2) unload as many bikes as possible to each bike shortage node (i.e., a station with not enough bikes to serve the cycling demand there) until the unmet demand at that node is eliminated or all bikes on the vehicle are unloaded.

The “effective” stage length of a stage in Step 2.2.3 is used to determine the route of a stage because the vehicle may not reach any node sharply at *l* time units. The effective stage length is defined as (ρ-ε) time units, where ε is the time of the vehicle arrived at the first node of the stage.

Figures 2 and 3 show the procedure related to Steps 2.2.2-2.2.4. For any stage σ (which is not the last stage), the operation time in the stage does not reach the “effective” stage length, the next node on the route z_y is added to the route z'_y and re-calculate the loading and unloading quantities (p'_{i_m} , r'_{i_m}). This process is repeated until the operation time exceeds the “effective” stage length. The solution (z'_y , p'_{i_m} , r'_{i_m}) of the last iteration becomes the solution of that run.

When the current stage is the final stage in the whole modeling horizon, the route z'_y needs to consider the total service time constraint in addition to effective stage length to maintain solution feasibility. As shown in Figure 3, nodes on the route z_y are added one by one to form a node sequence with the depot to be the final node, and the service time is then evaluated. When the route has not violated the total service time constraint, a new node is added to z'_y (in iteration (c_0 + 1)). If the newly added node violates the constraint, the solution (z'_y , p'_{i_m} , r'_{i_m}) in the last iteration becomes the solution of that run.

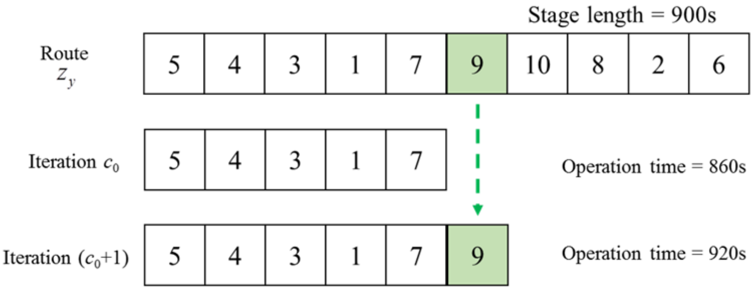

Figure 2 Mechanism of the iterated insertion of nodes in each stage except the final stage

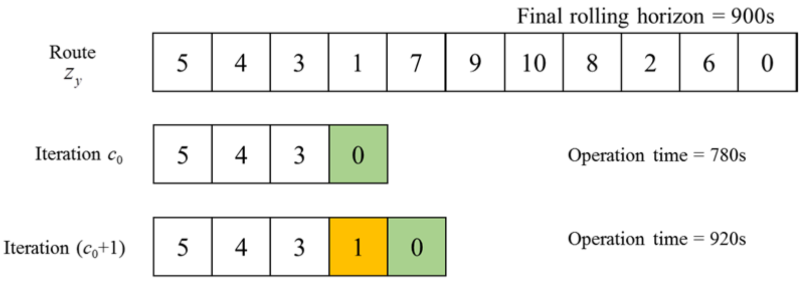

Figure 3 Mechanism of the iterated insertion of nodes in the final stage

According to Steps 2.2.2-2.2.4, each solution (z'_y, p'_{i_m}, r'_{i_m}) passed into Steps 2.2.5 and 2.2.6 consists of a feasible route (that satisfies the operation duration constraint) and the loading and unloading quantities of each visited node along the route. Step 2.2.5 determines the number of bikes on the vehicle between every pair of nodes (i.e., bike flows) along the route of stage τ based on the number of bikes on the vehicle at the beginning of this stage, $\hat{q}$, and the loading and unloading quantities of all nodes along the route, i.e., p'_{i_m} and r'_{i_m}. Mathematically, the bike flow between the (m-1)-th and the m-th nodes can be calculated by

$$\begin{aligned}\overline{q}_{i_{m-1}i_m} &= \overline{q}_{i_{m-2}i_{m-1}} + \left(p'_{i_{m-1}} - r'_{i_{m-1}}\right) = \overline{q}_{i_{m-3}i_{m-2}} + \left(p'_{i_{m-2}} - r'_{i_{m-2}}\right) + \left(p'_{i_{m-1}} - r'_{i_{m-1}}\right) = \dots \\ &= \hat{q} + \sum_{\gamma=1}^{m-1}\left(p'_{i_\gamma} - r'_{i_\gamma}\right).\end{aligned}$$

In Step 2.2.6, the unmet demand and the fuel and CO$_2$ emission cost are firstly determined based on the given loading and unloading quantities at each visited node and the bike flows along the route, respectively. The unmet demand of the m-th visited node along the route of stage τ can be determined by $\Lambda_{i_m} = \max\left[D'_{i_m} - \left(s'_{i_m} + r'_{i_m} - p'_{i_m} + \mu'_{i_m}\right), 0\right]$, where D'_{i_m} and s'_{i_m} respectively represent the expected cycling demand during that stage at node i_m and the number of bikes at the beginning of that stage at node i_m, and μ'_{i_m} is the total number of bikes returned to node i_m within that stage. For a non-visited node $j \in N \setminus G$, its unmet demand is determined in a similar way but without loading and unloading quantities, i.e., $\Lambda'_j = \max\left[D'_j - \left(s'_j + \mu'_j\right), 0\right]$. The fuel and CO$_2$ emission cost between the (m-1)-th node and the m-th node along the route of this stage (i.e., τ) is determined by $\overline{e}_{i_{m-1}i_m} = f_{\mathrm{C}}\kappa_1 d_{i_{m-1}i_m} / v_{i_{m-1}i_m,\tau}\left(hZV + w'\chi_1\phi_{1,i_{m-1}i_m} v_{i_{m-1}i_m,\tau} + \varphi\overline{q}_{i_{m-1}i_m}\chi_1\phi_{1,i_{m-1}i_m} v_{i_{m-1}i_m,\tau} + \phi_2\chi_1 v^3_{i_{m-1}i_m,\tau}\right)$, where

$v_{i_{m-1}i_m,\tau}$ is the average vehicle speed between the (m-1)-th node and the m-th node of this stage. Then, the weighted sum of the unmet demand and the fuel and CO_2 emission cost of a stage, $\left\{\alpha\left(\sum_{j\in G}\Lambda_j+\sum_{j\in N\setminus G}\Lambda'_j\right)+\sum_{\gamma=1}^{X-1}\overline{e}_{i_\gamma i_{\gamma+1}}\right\}$, can be calculated, and finally the fitness, which is the reciprocal of the above weighted sum, can be obtained.

3.5. The loading and unloading heuristic for the loading and unloading sub-problem

The loading and unloading sub-problem determines the loading and unloading quantities (p'_{i_m}, r'_{i_m}) of each visited node on a given route in a given stage. To solve this sub-problem, the simplest and fastest way is to adopt the greedy loading and unloading approach: 1) load as many bikes as possible at each bike surplus node onto the vehicle until the vehicle is full or all bikes at that node are loaded, and 2) unload as many bikes as possible to each bike shortage node until the unmet demand at that node is eliminated or all bikes on the vehicle are unloaded. However, the loading quantity at a bike surplus node depends on not only the node condition itself and the vehicle capacity but also the bike deficits of the drop-off nodes along the rest of the vehicle route. This greedy loading approach may let the vehicle load excessive amount of bikes and thus increase the fuel and CO_2 emission cost. This paper, therefore, introduces a novel and simple loading and unloading heuristic to determine the loading and unloading quantities at each node for a given route in each period. The objectives of this approach are two-folded: (1) avoid loading excess bikes (that are useless in solving unmet demand but increase the fuel and CO_2 emission cost); and (2) reduce the total distance travelled by the loaded bikes (which can hence reduce the fuel and CO_2 emission cost).

In this heuristic, we made use of the characteristics of pickup and drop-off nodes. The depot can be a drop-off and pickup node. The depot is a drop-off node only in the final stage for bike drop-off at the end of the repositioning operation, while it is a pickup node in other stages. We also define u_m as the total bike deficits of all nodes after the m-th node along the given route that can be satisfied by loading bikes at bike surplus nodes at or before this node. In other words, when u_m is positive, the total bike deficits of all nodes after the m-th node can only be solved by loading bikes at bike surplus nodes at or before the m-th node. The procedure of this heuristic is shown in the following steps.

Step 2.2.3.1: Set the current node position $m = X$ (i.e., route length) and $u_X = 0$.

Step 2.2.3.2: Determine u_m for $m = 2,...,X$ and the loading quantities of all bike surplus nodes:

(a) If the m-th node on the route is a drop-off node, set $u_{m-1} = \min\left(Q, u_m + a_m\right)$ and p'_{i_m} = 0, where a_m is the bike deficit of the m-th node. (The vehicle capacity term is used inside the minimum operator as bike deficits that exceed vehicle capacity cannot be solved anyway).

(b) If the m-th node on the route is a pickup node, set $u_{m-1} = \max\left(u_m - \beta_m, 0\right)$, where β_m is the bike surplus of the m-th node. (The zero term is used inside the maximum operator as u_m is always non-negative.) Meanwhile, the *initial* pickup quantity at this pickup node p''_{i_m} is calculated by $p''_{i_m} = \min\left(u_m, \beta_m\right)$. (The term u_m is included to ensure that all bikes loaded onto the vehicle at the m-th node are unloaded in subsequent nodes.)

(c) Set $m = m - 1$ and repeat the above steps until $m = 2$.

Step 2.2.3.3: Determine the loading and unloading quantities at the first node:

(a) In stage 1 (when the first node of the route must be the depot), the pickup quantity at the depot is equal to u_1, i.e., $p'_{i_1} = u_1$ (because the number of bikes loaded to the vehicle at the depot must not be greater than the vehicle capacity Q), and $r'_{i_1} = 0$.

(b) In subsequent stages, if the node is a pickup node, the loading quantity is determined by $p'_{i_1} = \min\left(\beta_1, \max\left(u_1 - \hat{q}, 0\right)\right)$ and $r'_{i_1} = 0$. If the node is a drop-off node, the unloading quantity is calculated by $r'_{i_1} = \min\left(a_1, \hat{q}\right)$ and $p'_{i_1} = 0$.

Step 2.2.3.4: Adjust the loading quantities of all subsequent pickup nodes:

(a) Set $F'_B = \hat{q} + p'_{i_1} - r'_{i_1}$ and $m = 2$.

(b) If the m-th node is a drop-off node, $m = m + 1$.

(c) If the m-th node is a pickup node, the loading quantity is obtained by $p'_{i_m} = \max\left(p''_{i_m} - F'_B, 0\right)$. Then, update F'_B by $F'_B = F'_B - \left(p''_{i_m} - p'_{i_m}\right)$ and $m = m + 1$.

(d) Repeat (b)-(c) until $m = X$.

Step 2.2.3.5: Determine the unloading quantities of all subsequent drop-off nodes:

(a) Set $F''_B = \hat{q} + p'_{i_1} - r'_{i_1}$ and $m = 2$.

(b) If the m-th node on the route is a pickup node, F''_B is updated by $F''_B = F''_B + p'_{i_m}$.

(c) If the m-th node on the route is a drop-off node, the unloading quantity at that node is calculated by $r'_{i_m} = \min\left(F''_B, a_m\right)$, and then update $F''_B = F''_B - r'_{i_m}$.

(d) Set $m = m + 1$. If $m \le X$, go to (b).

3.6. Neighborhood operators

Neighborhood operators are used to obtain new routing strategies by altering the positions of different bike nodes on a route z_y. When a new route is required in any phase of the EABC algorithm, a neighborhood operator is randomly chosen from a set of pre-selected neighborhood operators and applied once to the current route. In this study, we adopt three operators introduced by Szeto et al. (2011), including random swaps, subsequence reverse, and random swaps of reversed subsequence.

4. Numerical studies

We conducted computational experiments to (1) show the effectiveness of using the EABC algorithm compared with Genetic Algorithm to solve the routing sub-problem, (2) illustrate the effects of the weight α on the two objective values in different lengths of the modeling horizon, (3) demonstrate the effect of the stage length towards the objective values, and (4) investigate the effect of different loading and unloading times towards the objective values. The proposed solution method was coded in Visual C++ 2010 and ran on a computer with an Intel® Core™ i5 CPU1.6 GHz PC with a 4 GB RAM.

The experiments were performed using the sample networks adopted by Kloimüllner et al. (2014). However, in those instances, only the travel times but not the average vehicle speeds of all periods are given. As the fuel and CO_2 emission cost depends on both the travel times and average vehicle speeds of the links, this study assumes that the average vehicle speeds of all links are identical in the first stage (i.e., τ = 1 and $v_{ij,1} = 5$ m/s), and thus the average vehicle speed of link (i, j) in subsequent stages, $v_{ij,\tau}$, can be deduced by $v_{ij,\tau} = v_{ij,1} \cdot b_{ij,1} / b_{ij,\tau}$, where $b_{ij,1}$ and $b_{ij,\tau}$ are the travel times of link (i, j) in the first stage and the τ-th stage, respectively. Meanwhile, it is assumed that the average speeds of the links change every 4 hours. Also, the target demand level at each node in each stage is defined, unlike the target demand level remains unchanged throughout the whole operation in the study of Kloimüllner et al. (2014). The experiments worked on networks with sizes varying from 30 to 180 with one repositioning vehicle. The demand and number of returned bikes of a node were updated per hour. All the instances are available from http://web.hku.hk/~ceszeto/ShuiSzeto_DBGRP_data.zip. Both the loading time and unloading time of each bike are set to be 30 seconds (i.e., 0.5 minute) unless stated otherwise. For the parameter setting of the EABC algorithm, after considering the trade-off between computation time and solution quality in the preliminary experiments, the bee colony size was set to be 50 and the numbers of employed bees and onlooker bees were equal (i.e., 25 for each). The maximum cycle was set to be 200 times of the network size per stage and the value of '*Limit*' was fixed to 20 times of network size (i.e., the number of nodes). The stage length (except the last stage length) was set to be 2*l* (i.e., 2 times of the roll period) unless specified otherwise, and the length of each roll period *l* is set to be 60 minutes. All the test instances were run for 20 times.

4.1. Effectiveness of the EABC algorithm compared with Genetic Algorithm

To justify the use of the EABC algorithm for solving the routing sub-problem in the proposed solution method, the EABC algorithm was compared with the Genetic Algorithm (GA) introduced and investigated by Holland (1975). The whole process of the adopted GA approach is described as follows. In each iteration of GA, it involves crossover and mutation processes. The crossover process aims to generate a new set of routes (i.e., offspring) by the existing routes (i.e., parents). In this process, the parents are selected by the roulette wheel selection method according to their fitness, and then the selected parents then undergo the information exchange and create offspring. The selection and exchange processes are repeated until a targeted number of offspring have been generated. The mutation process is to introduce a degree of randomness in the pool of candidate routes, in which the information within each selected route is altered irreversibly. In this study, the mutation process is to apply one of the neighborhood operators introduced in Section 3.6 to a randomly selected route. After crossover and mutation, the parents and offspring are then evaluated. The better routes are left for the next iteration as the parents and the remaining (worse) routes are abandoned. The crossover and mutation processes become the two major components in the GA. A stopping criterion (generally the maximum number of iterations) is applied to terminate the algorithm.

The Genetic Algorithm here was coded by the authors. The route representation follows the work by van Breedam (1996), which used GA to solve classic vehicle routing problems. This representation is the same as that for the EABC algorithm coded in this study.

The GA and EABC algorithm were compared using 45 instances, which have different network sizes (from 30 to 180 nodes), lengths of modeling horizon (i.e., 6|T|, 12|T|, and 18|T|), and values of α. To have a fair comparison between the proposed method and the GA, both methods adopted the same set of neighborhood operators, had the same population size, and had approximately the same computation time for route search. The crossover and mutation rates of the GA were calibrated preliminarily and the rates of 0.5 and 0.05 were found to be the best choice in this problem, respectively. The average and best results of the proposed method and the GA are summarized in Table 4. For every instance, it can be named according to its network size |N|, its total number of periods over the time horizon T, and the value of the weight α. For the computation times of all instances, they are set to be (approximately) equal to the average computation times of the 20 runs of the EABC algorithm.

Table 4 demonstrates that the proposed EABC method outperforms the GA in all 45 instances, in which the average and minimum objective values obtained in the proposed method are lower than the corresponding values obtained by the GA. Moreover, the p-values of the t-test on the difference between the mean objective values generated by the two algorithms in all 45 instances are smaller than 0.01, which shows that the proposed method has statistically significant improvement over the GA in all the instances considered. To conclude, the test results show that the EABC algorithm is better than the GA as the sub-algorithm in the rolling horizon framework in solving our DGBRP.

Table 4 A comparison of the experimental results between the EABC and GA-based rolling horizon methods

Instances			EABC algorithm		Genetic algorithm		CPU time [c]	Imp%[d]	P-Value
|*N*|	*T*	*α*	Minimum[a]	Average[b]	Minimum[a]	Average[b]			
30	6	0.001	380.3	381.2	390.7	403.8	7.0	5.94	2.05E-06
		1	550.2	550.3	550.2	561.8	7.0	2.09	5.76E-03
		1000	109407.4	134803.9	154387.0	163896.9	7.0	21.58	4.46E-03
	12	0.001	776.2	779.7	794.7	812.2	14.0	4.16	2.59E-06
		1	950.3	951.9	974.8	996.1	14.0	4.64	2.30E-05
		1000	124805.7	147112.1	182813.2	191889.9	14.0	30.44	6.08E-03
	18	0.001	1165.9	1166.7	1209.9	1242.9	21.0	6.53	2.16E-07
		1	1394.8	1424.3	1439.4	1492.8	21.0	4.81	2.00E-05
		1000	161223.1	191802.2	212210.6	239671.1	21.0	24.96	1.30E-04
60	6	0.001	355.6	386.5	394.1	410.7	13.0	6.28	2.72E-04
		1	730.9	759.7	764.8	778.5	13.0	2.48	8.74E-05
		1000	343390.2	358409.2	368380.0	379790.1	13.0	5.97	2.44E-03
	12	0.001	756.1	766.6	791.0	809.2	27.0	5.57	2.94E-06
		1	1179.1	1206.1	1196.6	1223.9	27.0	1.48	2.68E-04
		1000	386812.7	404406.4	416756.4	438074.7	27.0	8.33	1.86E-04
	18	0.001	1112.0	1133.1	1162.7	1204.9	40.0	6.33	1.52E-06
		1	1726.8	1749.0	1784.9	1802.6	40.0	3.06	4.23E-08
		1000	518164.3	550070.4	613121.2	626220.9	40.0	13.84	2.99E-06
90	6	0.001	366.2	371.1	385.3	397.0	23.0	6.98	9.19E-07
		1	927.2	934.0	925.9	945.8	23.0	1.27	5.92E-03
		1000	512381.5	547693.9	553386.7	565378.3	23.0	3.23	7.47E-03
	12	0.001	752.0	757.3	773.9	795.2	47.0	5.01	2.72E-05
		1	1472.9	1482.8	1476.9	1518.3	47.0	2.39	6.64E-04
		1000	662792.9	685377.6	705747.1	715649.4	47.0	4.42	6.66E-07
	18	0.001	1107.1	1117.6	1148.8	1175.0	70.0	5.14	5.38E-06
		1	2042.2	2066.1	2104.1	2130.3	70.0	3.11	2.91E-06
		1000	851127.9	894455.2	851127.9	926109.8	70.0	3.54	3.20E-03
120	6	0.001	346.8	374.7	374.8	400.2	33.0	6.80	1.16E-03
		1	1227.0	1232.9	1227.5	1247.9	33.0	1.21	3.56E-03
		1000	812396.2	834607.9	833397.6	847577.2	33.0	1.55	4.72E-03
	12	0.001	736.7	748.1	780.6	793.1	67.0	6.02	7.18E-08
		1	1846.9	1860.4	1886.0	1905.8	67.0	2.44	7.55E-06
		1000	1004781.7	1020773.1	1076753.1	1094133.7	67.0	7.19	4.94E-09
	18	0.001	1114.8	1125.7	1164.9	1202.1	100.0	6.78	4.71E-06
		1	2523.1	2556.6	2591.0	2625.4	100.0	2.69	8.49E-07
		1000	1336167.4	1368564.4	1347104.2	1438601.0	100.0	5.12	1.19E-05
180	6	0.001	379.6	382.9	385.6	412.8	60.0	7.80	2.30E-05
		1	1543.6	1552.6	1564.7	1582.1	60.0	1.90	1.23E-04
		1000	988391.4	1002896.7	1154382.3	1172786.5	60.0	16.94	3.31E-12
	12	0.001	749.8	761.4	798.1	827.4	120.0	8.66	1.53E-06
		1	2122.9	2140.7	2195.4	2223.0	120.0	3.85	8.20E-07
		1000	1265789.6	1299890.4	1359726.0	1398552.4	120.0	7.59	6.07E-07
	18	0.001	1104.5	1130.8	1201.2	1233.9	180.0	9.12	9.78E-08
		1	3408.3	3431.4	3495.7	3551.5	180.0	3.50	1.07E-06
		1000	2178154.7	2226582.7	2248091.4	2292298.5	180.0	2.95	1.31E-05

[a] Average objective value obtained in 20 runs
[b] Minimum objective value obtained in 20 runs
[c] Average computation time obtained in 20 runs (in seconds)
[d] Calculated based on the mean values, referred to the improvement percentage against the GA-based method

4.2. *Effect of the weight α towards the total fuel and CO_2 emission cost and total unmet demand*

Due to the multi-objective nature of the DGBRP, the setting of the weight *α* for total unmet demand is important to achieve a balance between total unmet demand and the total fuel and CO_2 emission cost of the vehicle. Replicates of the algorithm with different weights were run and each replicate was run 20 times. In these experiments, the value of α varied from 0.0001 to 1000. This section presents the results for $H = 3|T|$, $6|T|$, and $12|T|$ with network size $|N| = 30$ for illustrative purposes.

Figure 4 illustrates the computational results of the experiments for $H = 3|T|$. It can be seen that the two objectives, the total unmet demand and the fuel and CO_2 emission cost per km, are conflicting. When α is smaller than 1, the algorithm considers the fuel and CO_2 emission cost to be extremely important. The algorithm then gives the largest total unmet demand and the lowest total fuel and CO_2 emission cost among all replicates. When α is greater than 1, the total fuel and CO_2 emission cost increase and the total unmet demand decreases simultaneously with respect to the increase of α. When α is greater than 100, the algorithm achieves the smallest total unmet demand while the total fuel and CO_2 emission cost still increases. A similar tradeoff is observed in Figure 5 and Figure 6. The total fuel and CO_2 emission cost reach the minimum when α was set to be very small values whereas total unmet demand is minimized when α was set to large values. It is noted that when the modeling horizon is longer, total unmet demand starts to drop with a smaller value of α but stops reducing at the same α value, i.e., 100. Meanwhile, the total fuel and CO_2 emission cost has slight fluctuations when α is smaller than 1 and keeps increasing when α is greater than 1. In all cases, when the weight for total unmet demand increases, the total fuel and CO_2 emission cost becomes less important. The repositioning strategy therefore aims to handle more unmet demand by loading more bikes on the vehicle to transport and serve bike deficit nodes, and as a result increases the average vehicle load and the total fuel and CO_2 emission cost.

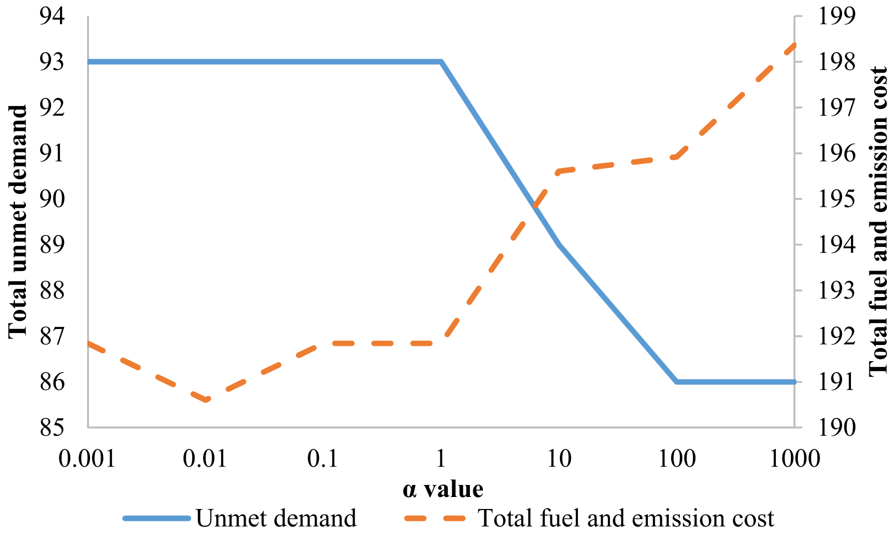

Figure 4 Effect of the weight α at $H = 3|T|$

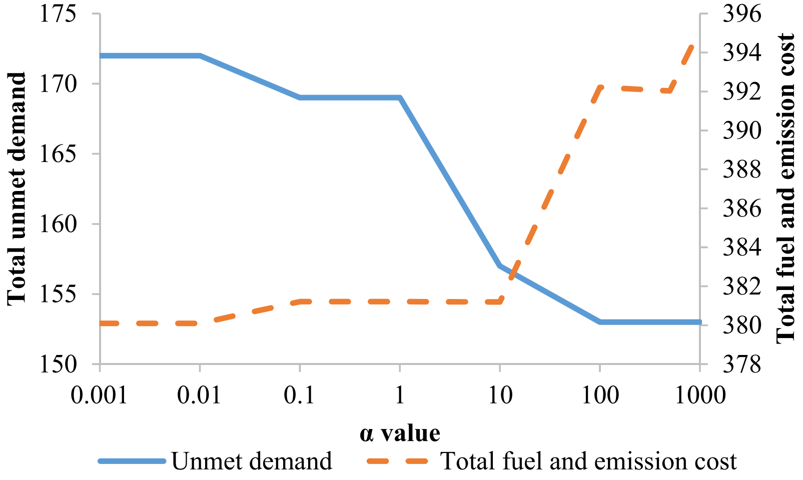

Figure 5 Effect of the weight α at $H = 6|T|$

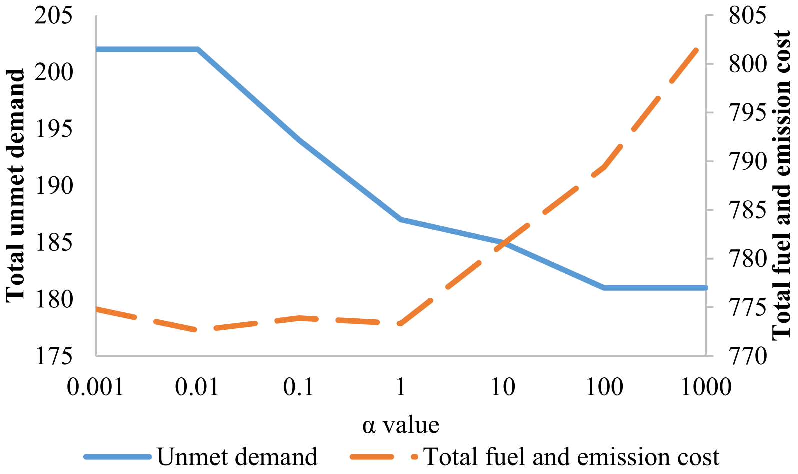

Figure 6 Effect of the weight α when $H = 12|T|$

4.3. Effect of the stage length towards the total fuel and CO_2 emission cost and total unmet demand

This section investigates how the stage length influences the total fuel and CO_2 emission cost and total unmet demand. The stage length ρ can be $2l$, $3l$, and $4l$ and the total operation time $|H|$ is set to be $6l$. The 30-node network was used and α ranges from 0.01 to 1000.

Figure 7 depicts the total fuel and CO_2 emission cost with stage lengths to be $2l$, $3l$, and $4l$ under different values of α. The trends of all cases are similar that the fuel and CO_2 emission cost increases with the value of α. For the case of $\rho = 4l$, its total fuel and CO_2 emission cost is higher than that in the other two cases for all α values. With a longer stage length, the vehicle may tend to plan for a route to load more bikes at the earlier stages to solve the bike deficits of a series of bike deficit nodes, but not to visit the bike surplus and deficit nodes alternately. This results in a larger bike load on the vehicle and thus higher fuel and CO_2 emission cost. For the cases of $\rho = 2l$ and $\rho = 3l$, their difference in the total cost are small when the α value is smaller than 10. However, the total cost of the case of $\rho = 3l$ increases sharply when the α value is greater than 10, while the increase in fuel and CO_2 emission cost of the case of $\rho = 2l$ is less steep. In other words, the total fuel and CO_2 emission cost of the case of $\rho = 2l$ is the lowest when the α value is greater than 10. For the total unmet demands shown in Figure 8, the differences among all three cases are not significant when the α value is smaller than 1. However, for all α values greater than 1, the case of $\rho = 2l$ achieves the lowest total unmet demand compared with the other two cases. This implies that for larger α values, a longer stage period can result in higher total unmet demand (and the gap can be as large as 13 bikes at $\alpha = 1000$ between the cases of $\rho = 2l$ and $\rho = 4l$). This difference may be due to prediction accuracy. Planning a route for a longer period can create larger total unmet demand because the demand forecasts for later time intervals are available with lower reliability.

An additional note related to the stage length is about the computation times. Table 5 displays the average computation times for 20 runs for all three different stage lengths with α = {0.01, 1, 1000}. The computation times do not have observable differences with respect to the α value (as the largest gap is less than 0.2s), but show an increasing trend with respect to the length of stage period ρ. With a longer stage length, the route is longer and thus requires more time to evaluate the loading and unloading quantities at each node.

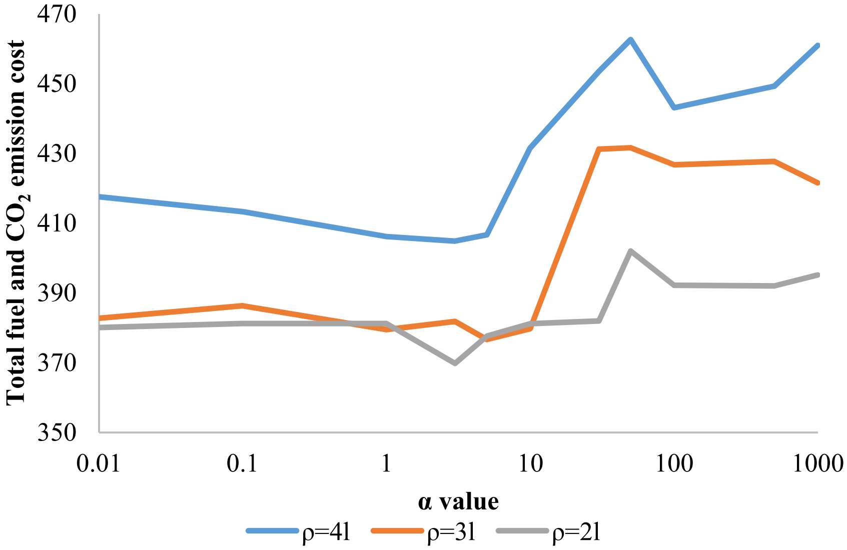

Figure 7 Effect of stage length on the total vehicle fuel and CO_2 emission cost

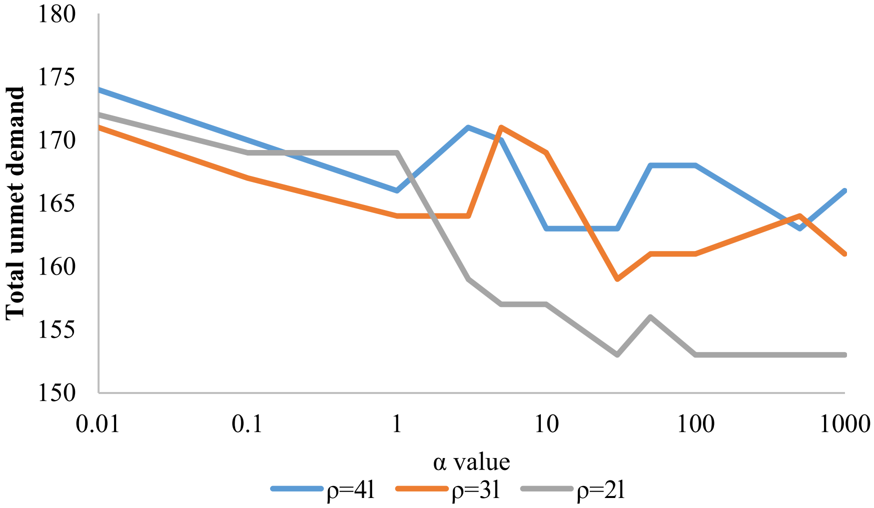

Figure 8 Effect of stage length on total unmet demand

Table 5 Computation times under different combinations of stage lengths and α values

α value	$\rho = 2l$	$\rho = 3l$	$\rho = 4l$
0.01	8.89	9.28	9.76
1	8.76	9.18	9.81
1000	8.83	9.13	9.85

4.4. Effect of the loading and unloading time towards the total fuel and CO_2 emission cost and total unmet demand

The loading and unloading times of a bike vary from one public bike sharing system to another. Though the deviation of loading and unloading times per bike may not be very large, the accumulated time deviation can be significant when the service time horizon is long. This section compares how the lengths of these times influence total unmet demand and the total fuel and CO_2 emission cost per unit travel time. This example assumes that both loading and unloading time are equal and considers the loading time (and unloading time) to be 6, 30, and 60 seconds per bike, with $H = 6|T|$ and $|N| = 30$.

Figure 9 shows the effect of loading time on total unmet demand. For small values of α, the differences in loading time do not influence total unmet demand because minimizing the total fuel and CO_2 emission cost is the dominant objective. When α increases, we can observe that a shorter loading time can achieve a lower total unmet demand. A shorter loading time per bike allows more time for travel within a roll period, and thus the vehicle can visit more nodes to satisfy their demand.

Figure 10 compares the total emissions per unit travel time under different loading and unloading times. This normalization of the fuel and CO_2 emission cost is required because the vehicle travel times of the three instances have large deviations despite the same length of modeling horizon. The total travel time occupies approximately 96% of the modeling horizon when the loading time is 6 seconds, whereas it only occupies 68–72% when the loading time is 60 seconds. As it is assumed that there is no fuel consumption and CO_2 emission during the loading and unloading periods, the total fuel and CO_2 emission cost in all instances should be normalized by the total travel time to obtain the total fuel and CO_2 emission cost per unit time (and hence the emission rate). The results show that the higher loading and unloading times always give a higher total fuel and CO_2 emission cost per unit time. It is because a longer loading time results in visiting fewer nodes, and thus more bikes need to be loaded onto vehicles when visiting a bike surplus node. These results also indicate that shortening loading and unloading times is an effective way to reduce total unmet demand and the emission cost per unit time of the vehicle.

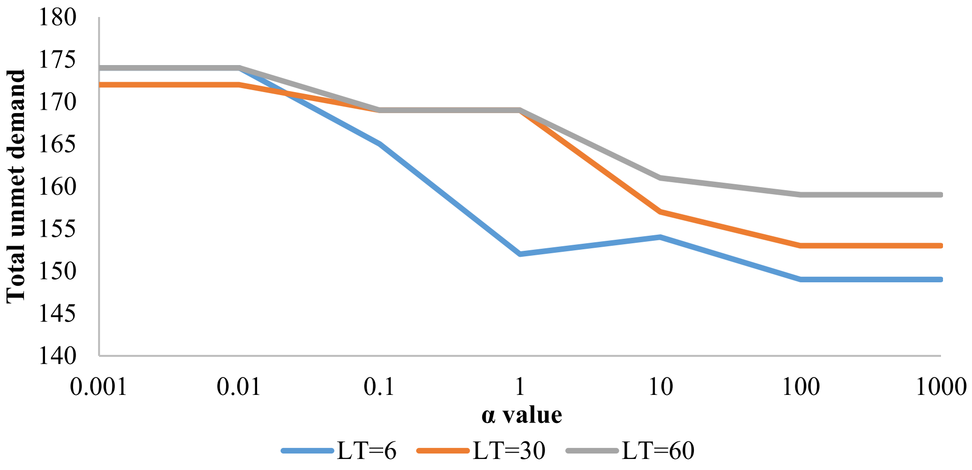

Figure 9 Effect of loading time on total unmet demand

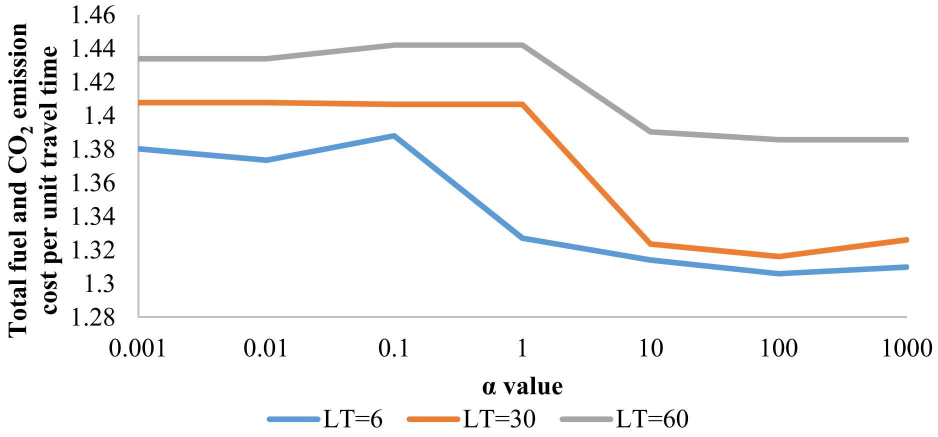

Figure 10 Effect of loading time on the total fuel and CO_2 emission cost per km

5. Conclusions

We propose a novel dynamic green bike repositioning problem in a bike-sharing system. The problem aims to reduce the total unmet demand of the bike-sharing system and total fuel and CO_2 emission cost of the repositioning vehicle over a specific service time horizon. To handle the dynamic nature of the problem, this study adopts a rolling horizon approach to break down the proposed problem into a set of stages, in which a static bike repositioning sub-problem is solved in each stage. An EABC algorithm and a route truncation heuristic are jointly used to optimize the route design in each stage, and the loading and unloading heuristic is used to tackle the loading and unloading sub-problem along the route in a given stage.

Numerical examples are set up to illustrate the performance of the proposed algorithm and the properties of the problem. The results show that the EABC algorithm outperforms the classic Genetic Algorithm to solve the routing sub-problem. The results also reveal that the shorter stage length can achieve a better solution; in addition to computation time, total unmet demand and the total fuel and CO_2 emission cost are higher with a longer stage length. Moreover, the results also demonstrate that the setting of weight is important for achieving a balance between the two objectives, i.e., minimizing total unmet demand and minimizing the total vehicle fuel and CO_2 emission cost, despite the length of the service time horizon. The observed tradeoff leaves the decision of the best balance between two objectives to the operator. In addition, to reduce total unmet demand and the total fuel and CO_2 emission cost per unit time in practice, the numerical results show that one of the solutions is to shorten loading and unloading times per bike.

To the best of our knowledge, the branch and price method has not been used to solve bike repositioning problems, including ours. This can be left for future research.

Acknowledgements

This research was jointly supported by a grant (No. 201511159095) from the University Research Committee of the University of Hong Kong, and a grant from National Natural Science Foundation of China (No. 71271183). The authors are grateful to the two reviewers for their constructive comments. The authors are also grateful for the three guest editors for their invitation to us to submit a paper to this special issue.

References

Alvarado-Iniesta, A., Garcia-Alcaraz, J. L., Rodriguez-Borbon, M. I., & Maldonado, A. (2013). Optimization of the material flow in a manufacturing plant by use of artificial bee colony algorithm. *Expert Systems with Applications*, *40*(12), 4785-4790.

AMITRAN (2016). *Emission Model Selection*. Amitran Information and Guidance Knowledge Base. Retrieved October 05, 2016, from <http://amitran.teamnet.ro/index.php/Emission_model_selection>

Barth, M., Younglove, T., & Scora, G. (2005). Development of a heavy-duty diesel modal emissions and fuel consumption model. *Technical Report*. UC Berkeley: California Partners for Advanced Transit and Highways (PATH), California, USA. Retrieved from <http://www.path.berkeley.edu/PATH/Publications/PDF/PRR/2005/PRR-2005-01%.pdf>

Bektaş, T., & Laporte, G. (2011). The pollution-routing problem. *Transportation Research*

Part B: Methodological, *45*(8), 1232-1250.

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-Operations Research*, *45*(1), 37-61.

Caggiani, L., & Ottomanelli, M. (2012). A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia - Social and Behavioral Sciences*, *54*, 675-684.

Chemla, D., Meunier, F., & Wolfler Calvo, R. (2013). Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*, *10*(2), 120-146.

Contardo, C., Morency, C., & Rousseau, L. M. (2012). Balancing a dynamic public bike-sharing system. *Technical Report CIRRELT-2012-09*, Montréal.

Cordeau, J. F., Dell’Amico, M., Falavigna, S., & Iori, M. (2015). A rolling horizon algorithm for auto-carrier transportation. *Transportation Research Part B: Methodological*, *76*, 68-80.

DEFRA. (2012). 2012 Guidelines to DEFRA/ DECC’s GHG conversion factors for company reporting: Methodology paper for emission factors. *Technical Report*, London, United Kingdom. Retrieved from <http://https://www.gov.uk/government/uploads/system/uploads/attachment_data/fi%le/69568/pb13792-emission-factormethodology-paper-120706.pdf>

Dell'Amico, M., Hadjicostantinou, E., Iori, M., & Novellani, S. (2014). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, *45*, 7-19.

Dell'Amico, M., Iori, M., Novellani, S., and Stützle, T., 2016. A destroy and repair algorithm for the bike sharing rebalancing problem. *Computers and Operations Research*, *71*, 149-162.

Demir, E., Bektas, T., & Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, *6*(5), 347–357.

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*(2), 346-359.

Demir, E., Bektaş, T., & Laporte, G. (2014). A review of recent research on green road freight transportation. *European Journal of Operational Research*, *237*(3), 775-793.

Department of Information and Tourism, Taipei City Government. (2013). *YouBike Be Better*. Retrieved from Department of Information and Tourism, Taipei City Government: <http://reading.gov.taipei/ct.asp?xItem=68240488&CtNode=69314&mp=100021>

Di Gaspero, L., Rendl, A., & Urli, T. (2013). Constraint-based approaches for balancing bike sharing systems. *Principles and Practice of Constraint Programming*, 758-773.

E3M Lab. (2014). *Manual of the PRIMES-TREMOVE Transport Model*. Retrieved October 12, 2016, from <http://147.102.23.135/e3mlab/PRIMES%20Manual/The%20PRIMES-TREMOVE%20MODEL%202013-2014.pdf>

Erdoğan, G., Laporte, G., & Wolfler Calvo, R. (2014). The static bicycle relocation problem with demand intervals. *European Journal of Operational Research*, *238*(2), 451-457.

Ericsson, E., Larsson, H., & Brundell-Freij, K. (2006). Optimizing route choice for lowest fuel consumption–potential effects of a new driver support tool. *Transportation Research Part C: Emerging Technologies*, *14*(6), 369-383.

European Environment Authority. (2016). EMEP/EEA air pollutant emission inventory guidebook 2016. *EEA Report No 21/2016*. European Environment Authority. Retrieved from <http://www.eea.europa.eu/publications/emep-eea-guidebook-2016/>

Figliozzi, M. (2010). Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board*, *2197*, 1-7.

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

GHG Protocol (2013). GHG emissions from transport or mobile sources. *Technical Report,* Greenhouse Gas Protocol, USA. Retrieved from <http://www.ghgprotocol.org/calculation-tools/all-tools>

Hausberger, S., Rexeis, M., Zallinger, M., & Luz, R. (2009). Emission factors from the Model PHEM for the HBEFA: Version 3. *Technical Report*, TU Graz, Austria.

Hickman, J., Hassel, D., Joumard, R., Samaras, Z., & Sorenson, S. (1999). Methodology for calculating transport emissions and energy consumption. European Commission, DG VII.

Technical Report. Retrieved from <http://www.transport-research.info/Upload/Documents/200310/meet.pdf>

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

Ho, S. C., & Szeto, W. Y. (2016). GRASP with path relinking for the selective pickup and delivery problem. *Expert Systems with Applications*, *51*(1), 14-25.

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

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press.

Jabali, O., Woensel, T., & de Kok, A. G. (2012). Analysis of travel times and CO2 emissions in time‐dependent vehicle routing. *Production and Operations Management*, *21*(6), 1060-1074.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report TR06*, Erciyes University; 2005.

Karaboga, D., & Gorkemli, B. (2011). A combinatorial artificial bee colony algorithm for traveling salesman problem. *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 50-53.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute*, *346*(4), 328-348.

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*, Volume 8600 of the series Lecture Notes in Computer Science, 73-84. Springer Berlin Heidelberg.

Koç, Ç., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, *70*, 239-254.

Kopfer, H. W., Schönberger, J., & Kopfer, H. (2014). Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. *Flexible Services and Manufacturing Journal*, *26*(1-2), 221-248.

Li, J. Q., Pan, Q. K., & Gao, K. Z. (2011). Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. *The International Journal of Advanced Manufacturing Technology*, *55*(9-12), 1159-1169.

Li, Y., Szeto, W. Y., Long, J., & Shui, C. S. (2016). A multiple type bike repositioning problem. *Transportation Research Part B: Methodological*, *90*, 263-278.

Lin, J. H., & Chou, T. C. (2012). A geo-aware and VRP-based public bicycle redistribution system. *International Journal of Vehicular Technology*, *2012*, 1-14.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, *7*(4), 326-329.

Nair, R., Miller-Hooks, E., Hampshire, R. C., & Bušić, A. (2013). Large-scale vehicle sharing systems: Analysis of *Vélib'*. *International Journal of Sustainable Transportation*, *7*(1), 85-106.

Ntziachristos, L., Gkatzoflias, D., Kouridis, C., & Samaras, Z. (2009). COPERT: a European road transport emission inventory model. In *Information Technologies in Environmental Engineering*, 491-504. Springer Berlin Heidelberg.

Pan, Q. K., Tasgetiren, M. F., Suganthan, P. N., & Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, *181*(12), 2455-2468.

Peeta, S., & Mahmassani, H. S. (1995). Multiple user classes real-time traffic assignment for online operations: A rolling horizon solution framework. *Transportation Research Part C: Emerging Technologies*, *3*(2), 83-98.

Raidl, G. R., Hu, B., Rainer-Harbach, M., & Papazek, P. (2013). Balancing bicycle sharing systems: Improving a VNS by efficiently determining optimal loading operations. *Hybrid Metaheuristics*, 130-143.

Rainer-Harbach, M., Papazek, P., Hu, B., & Raidl, G. R. (2013). Balancing bicycle sharing systems: A variable neighborhood search approach. *Evolutionary Computation in Combinatorial Optimization*, 121-132.

Rainer-Harbach, M., Papazek, P., Raidl, G. R., Hu, B., & Kloimüllner, C. (2015). PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems. *Journal of Global Optimization*, 1-33.

Raviv, T., Tzur, M., & Forma, I. A. (2013). Static repositioning in a bike-sharing system: Models and solution approaches. *EURO Journal on Transportation and Logistics*, *2*(3), 187-229.

Schuijbroek, J., Hampshire, R., and van Hoeve, W. J., 2017. Inventory rebalancing and vehicle routing in bike sharing systems, *European Journal of Operational Research*, *257*(3), 992–1004.

Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, *9*(2), 625-631.

Stolletz, R., & Zamorano, E. (2014). A rolling planning horizon heuristic for scheduling agents with different qualifications. *Transportation Research Part E: Logistics and Transportation Review*, *68*, 39-52.

Sundar, S., Singh, A., & Rossi, A. (2010). An artificial bee colony algorithm for the 0–1 multidimensional knapsack problem. *Contemporary Computing*, 141-151.

Szeto, W. Y., Wu, Y., & Ho, S. C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, *215*(1), 126-135.

Szeto, W. Y., Jaber, X., & Wong, S. C. (2012). Road network equilibrium approaches to environmental sustainability. *Transport Reviews*, *32*(4), 491-518.

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

Ting, C. K., & Liao, X. L. (2013). The selective pickup and delivery problem: Formulation and a memetic algorithm. *International Journal of Production Economics*, *141*(1), 199-211.

United States Environmental Protection Agency. (2010). *MOVES (Motor Vehicle Emission Simulator)*. Retrieved from <http://www.epa.gov/otaq/models/moves/index.htm>

van Breedam, A. (1996). An analysis of the effect of local improvement operators in genetic algorithms and simulated annealing for the vehicle routing problem. *RUCA working paper 96/14*, University of Antwerp, Belgium.

Vock, C., Hubel, J., Tsokolis, D., Samaras, C., Ntziachristos, L., Tola, R., Ricci, C., & Samaras, Z. (2014). Report on the development and use of the vehicle energy/ emission simulator. *Technical Report, v.1.3*. ICT-EMISSIONS Consortium. Retrieved from <http://www.ict-emissions.eu/wp-content/uploads/2012/05/D3.2.1-Report-Vehicle-Simulator_Final.pdf>

Wiersma, B. (2010). Bicycle sharing system: Role, effects and application to Plymouth. Groningen: Masters, University of Groningen. Retrieved from <https://www.rug.nl/research/portal/files/14446525/EES-2010-102M_BoukeWiersma.pdf>

Yao, B., Hu, P., Zhang, M., & Wang, S. (2013). Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem. *Simulation*, *89*(6), 762-770.

Zhang, S., Lee, C. K. M., Choy, K. L., Ho, W., & Ip, W. H. (2014). Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. *Transportation Research Part D: Transport and Environment*, *31*, 85-99.

Zhao, F., Li, S., Sun, J., & Mei, D. (2009). Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. *Computers and Industrial Engineering*, *56*(4), 1642-1648.