Original Article

Bike-sharing rebalancing problem:
Modified artificial bee colony algorithm modeling

Kanokporn Boonjubut (Graduate School of Engineering and Science, Shibaura Institute of Technology, nb19103@shibaura-it.ac.jp)
Hiroshi Hasegawa (Graduate School of Engineering and Science, Shibaura Institute of Technology, h-hase@shibaura-it.ac.jp)
Suphanut Kongwat (Department of Mechanical Engineering, King Mongkut's University of Technology Thonburi, suphanut.kon@mail.kmutt.ac.th)

Abstract
The number of bike-sharing services has rapidly increased in many cities worldwide. One of the main challenges of the bike-sharing system operation costs is allocating enough bikes and parking space. This paper presents a model for solving the bike-sharing relocation problem. The artificial bee colony (ABC) algorithm is an efficient approach, but it is still insufficient for the selection strategy. ABC has been adopted in various problems to improve the performance of various systems. This research proposed a modified ABC algorithm in a neighbor solution to enhance the solution performance, namely guided local search (GLS), to apply to the design route for transportation while truck relocation bikes each station in the bike-sharing system. Computational experiments were performed to find out the best modeling solution in the case. The implementations were experimental for the same data instances, which made it possible to compare the performance algorithms so as to solve the bike-sharing relocation problem of the pickup and drop off. The results showed that the GLS-ABC method can be a better solution than the original one. The statistically significant p-value of the mean objective value of the different algorithms was smaller than 0.05. Thus, the impact of minimizing the route tour cost in solving the bike-sharing relocation problem.

Keywords
vehicle routing, optimization, artificial bee colony algorithm, bike-sharing relocation problem, swarm intelligence

1. Introduction

Tourism has become increasingly essential worldwide. According to many statistics, the number of tourists tends to continuously increase, especially in Japan [Japan Tourism Statistics, 2020]. Furthermore, tourism is related to travel, so transportation is vital for moving passengers and goods from one point to another. Travelers have various cost expenses and satisfaction factors that impact their preferred modes of transportation, which include driving, walking, using a taxi, or using a bike-sharing service.

In response to saving time, common short travel distances, and multimodal transportation connections, bike-sharing services have made bike rentals available for travelers and tourists so that they can rent bikes and return them at any station. Most bike-sharing systems provide automatic systems for users and operators so that customers can just use their smartphones to locate available bike stations, which makes it easier and more convenient to attract more customers. Recently, the frequency of using bike-sharing services has significantly increased due to the popularity of green travel, as many people have become more aware of pollution and other common health issues. Moreover, due to the increase in carbon dioxide (CO_2) levels, many people are taking more measures to reduce greenhouse gas emissions in every economic sector.

However, with the increasing popularity of bike-sharing services, many problems have arisen, including problems during peak hours, not enough bikes at some stations, etc. Also, sometimes, there are no available parking spaces to return bikes, especially in residential areas or near train stations. Thus, the operators of bike-sharing systems face many challenges when it comes to the allocation of enough bikes and parking spaces. This problem can be solved by determining the route of relocation bikes, which includes the picked up and returned bikes. Thus, cost-efficient operation can be used to guarantee profit maximization optimization. Moreover, customer satisfaction is important so customers can continue using the service, and it is also important for increasing the number of customers and improving service reliability.

Many researchers have found that the artificial bee colony (ABC) algorithm can be used to effectively solve capacity vehicle routing problems [Szeto et al., 2011]. This paper presents a newly developed method for efficiently exploring an ABC algorithm to solve the bike-sharing relocation problem and reduce the operating costs along with determination of the minimum cost based on the vehicle routing distance while bike relocation operations maintain a sequence of visiting each station.

The rest of the paper is organized as follows. In Section 2, we explained other related works concerning the relocation problems of bike-sharing. Then, the main bike-sharing relocation problem was described in Section 3. We briefly explained the main algorithm for solving the bike-sharing relocation problem in Section 4. Then, the proposed algorithm for enhancing the ABC algorithm's exploitation capability in solving the bike-sharing relocation problem was described in Section 5. The results were discussed in Sections 6-7. Finally, the conclusions and future work were presented in Section 8.

2. Related work

Recently, bike-sharing systems have received a lot of attention from researchers. Operational relocation can be classified into two groups. The first group includes user-based strategies, which incentivize users to participate and encourage them to voluntarily relocate their rented bikes. Such strategies include static pricing and dynamic pricing strategies. Singla et al., [2015] presented a crowdsourcing mechanism for dynamic pricing, which enabled the calculation of each station's incentive values and the development of a dynamic incentives system by offering incentive amounts to users and utilizing smartphone applications. The second group includes operator-based strategies, where relocation operators work on optimizing the pickup and delivery costs. Erdoğan et al. [2015] presented an exact algorithm using a branch-and-cut algorithm which utilizes combinatorial Benders' cuts to separate infeasible solution from the feasible region to solve the static bicycle rebalancing problem by determining the minimum cost sequence of the stations to be visited by a single vehicle. Cruz et al. [2017] presented an iterated local search heuristic for solving the bicycle rebalancing problem. Gajpal and Abad [2009] proposed a construction rule in Ant Colony Optimization as two multi-route local searches to solve VRP with simultaneous delivery and pickup. Shui and Szeto [2017] offered a dynamic green bike repositioning problem that simultaneously minimizes the total unmet demand of bike-sharing systems and the fuel and CO_2 emission costs of repositioning vehicles. The solution method was based on the ABC algorithm.

Numerous studies on bike-sharing systems have proposed solutions for the rebalancing problem for operating a vehicle routing. Consequently, the underlying vehicle routing problem has received the most attention. In our study, we have proposed the effectiveness of the solution. We also modified the efficiency of the swarm-based metaheuristic algorithm, ABC, and enhanced its search efficiency in solving the bike-sharing relocation problem.

3. Bike-sharing relocation problem description

3.1 Problem description

Bike-sharing relocation is a part of VRP that can be applied to relocation. Battarra et al. [2014] classified the VRP as widely treated pickup and delivery problems (PDP). PDP is a kind of VRP where goods must be transported from different origins to different destinations, and it is divided into three classes. The first class is one-to-one (1-1), where each good and request are provided with a pair of origin and destination. The second class is the One-to-Many-to-One (1-M-1), which represents how some goods must be delivered from a depot to customers and how other goods must be recollected from the customers and carried back to the depot. The third class is Many-to-Many (M-M), which represents how each good can have various origins and destinations and how any location can be the origin and destination. The problem of relocating bikes in a bike-sharing system lies in this class.

The model for solving vehicle routing problems for deliveries and pickups (VRPDP) aims at minimizing the cost or distance by providing customers with the allocation of vehicle routes for truck trips to service customers. One limitation that must be considered is the vehicle loading capacity. Although this is a significant problem, it is not extending VRP due to the lack of multiple travel plans. Customers receiving goods from a depot are called linehaul or deliveries. Customers who send goods back are called pickup or backhauls. It is possible that customers may want to both send and receive goods at the same time. This case is called combined demands. Also, in bike-sharing relocation, a customer at a station may need to pick up or drop off a bike [Usama et al., 2020]; hence, it is possible to adopt VRPDP with combined demand in solving this problem. This can be modeled using integer linear programming models, which involve minimizing the cost or distance. Starting from the depot, the truck drives to visit each station exactly once to drop off or pick up a bike for relocation. Then, the truck must be back to the depot. The problem can be defined on a graph $G = (V, A)$, where $V = \{1...., n\}$ is a set of nodes (stations) in a depot located at node 0, and $A = \{(i, j): i, j \in V\ i \neq j\}$ is a set of arcs (distance between each pair of vertices). Each station i has a demand dd_i or pd_i, where pd_i denotes the pickup bikes that must be removed at station i, and dd_i denotes the drop off bikes that must be supplied at station i. The bikes removed from the pickup stations can either go to a drop off station or back to the depot. The bikes supplied to a drop off station can either come from the depot or from a pickup station. Also, the feet of m is the limitation of the capacity Q of each available vehicle at the depot. This problem is related to determining a relocation while minimizing the total cost of a fleet with a number of (m) vehicles through the graph.

There are various methods to solve bike-sharing relocation as a vehicle routing problem. The Swarm Intelligence is an efficient method as the ABC algorithm [Karaboga and Basturk, 2007], but it is still insufficient for the selection strategy. Koraboga and Basturk [2008] and Pathak et al. [2017] proposed the modified ABC by using Local Search (LS). The GLS is a way to improve the selection strategy in the ABC algorithm because the GLS gets the results better than LS [Kilby et al., 1999]. To narrow the gaps of the previous research work, this paper proposes the modified ABC in a neighbor solution to enhance the solution performance, namely GLS to improve the solution performance to apply in bike-sharing relocation problem as follows in Section 5.

3.2 Mathematical modeling for the bike-sharing problem

According to [Wassan and Nagy, 2014], with the objective of minimizing the total cost, a model was proposed for solving the VRPDP in order to make the mathematical model of the bike-sharing problem mimic VRPDP. The mathematical model for minimizing the total cost in this problem was defined as follows:

Parameter

D_{ij}: Distance between the station locations i and j

dd_i: Drop off demand of station i

pd_i: Pickup demand of station i

Q: Vehicle capacity (number of bikes)

m: Number of vehicles

Dr_{ij}: Number of drop off on board on the arc ij

Pu_{ij}: Number of pick up on board on the arc ij

Decision variables

$$x_{ij} = \begin{cases} 1, \textit{when vehicle m travels from station i to station j} \\ 0, \textit{otherwise.} \end{cases}$$

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} D_{ij} x_{ij} \tag{1}$$

$$s.t. \sum_{i=0}^{n} x_{ij} = 1, j \in \{1,...,n\} \tag{2}$$

$$\sum_{j=0}^{n} x_{ji} = 1, j \in \{1,...,n\} \tag{3}$$

$$\sum_{i=0}^{n} Dr_{ij} - dd_i = \sum_{i=0}^{n} Dr_{ji}, j \in \{1,...,n\} \tag{4}$$

$$\sum_{i=0}^{n} Pu_{ij} + pd_i = \sum_{i=0}^{n} Pu_{ji}, j \in \{1,...,n\} \tag{5}$$

$$\sum_{i=0}^{n} Dr_{i0} = 0 \tag{6}$$

$$\sum_{i=0}^{n} Pu_{0i} = 0 \tag{7}$$

$$Dr_{ij} + Pu_{ij} \le Qx_{ij}, i, j \in \{1,...,n\} \tag{8}$$

$$\sum_{i=1}^{n} x_{0i} = m \tag{9}$$

$$x_{ij} = \{0,1\}, i, j \in \{1,...,n\} \tag{10}$$

The objective function is to minimize the total travel distances of all trucks drivers for bike relocation that define the feasible solutions of the routes given the constraints. Constraints (2) and (3) ensure that the vehicle must only visit stations. Constraint (4) and (5) guarantee that the flow conservation constraints are met. Constraints (6) and (7) confirm that the vehicle starts at the depot with zero pickup bikes and finishes with zero drop off bikes. Constraint (8) make sure that the vehicle picks up and drops off loads at any customer location within the vehicle load capability. Constraint (9) verify that the vehicle leaves from the depot, and constraint (10) is a binary variable.

4. Trial and error of the algorithm for solving relocation bike-sharing

To solve the vehicle routing problem, researchers have proposed a variety of methods, such as the exact, heuristic, and metaheuristic methods. Mathematical models can be used to solve such problems and to explain various problem aspects. Likewise, metaheuristic methods, such as the genetic algorithm or ABC algorithm, have been applied to solve such problems.

The most common relocation operation problem that exists all day every day is the bike-sharing relocation operation problem, and it can either be static or dynamic. The static problem is when relocation is performed on a predetermined schedule when a system is closed or minimally operating at night. The dynamic problem is when relocation occurs in the daytime when the system rapidly changes and needs relocation. Essentially, for bike-sharing relocation systems in small or medium cities, the bikes are often carried out at night using a vehicle that visits each station exactly once. Many researchers proposed methods for solving the vehicle routing problem to find out the minimizing cost. Using the exact method (genetic algorithm (GA)) [Katoch et al., 2020], that has been widely used in various real-life applications, we used a trial-and-error experiment to compare the performances of the well-known in the literature and those of different kinds of optimization algorithms consisting of mixed intergenerational problems (MIP). The representation of chromosomes is closely associated with real-life problems. The main advantages of GA are that it is robust, efficient, and accurate, and the artificial bee algorithm (ABC) [Kumar et al., 2017; Okoro et al., 2019] has become one of the most common optimization methods in the field of artificial intelligence since it was first conceived in the early nineties. As a result, many research works elaborated on the value of using it in well placement optimization to solve the bike-sharing relocation problem. Gurobi solved such problems using MIP. GA and ABC were coded in Python.

In the trial-and-error experiments, the ABC algorithm showed the best performance, but it took a long time compared with the other methods. We aimed at modifying the

Table 1: Comparison of the performances of the experimental results for each algorithm

Number of stations		Algorithm		
		MIP	GA	ABC
10	Avg. (km.)	348.96	291.57	282.84
	CPU time (S.)	9.12	1.96	8.29
20	Avg.(km.)	303.29	313.15	302.05
	CPU time (S.)	10.23	2.8	10.25
30	Avg.(km.)	n/s	567.70	423.90
	CPU time (S.)	> 24hr.	2.98	20.56

performance of the ABC algorithm in solving the bike-sharing relocation problem. The results were consistent with those of Rothlauf [2011], who proposed an exact optimization method that guaranteed finding an optimal solution. Also, using heuristic optimization methods, there were no guarantees that an optimal solution can be found. Usually, the exact optimization method is a choice method if it can solve an optimization problem with an effort that polynomially grows with the problem size. The situation is different if the problems are NP-hard, as exact optimization methods need exponential effort. Then, even medium-sized problem instances often become intractable and cannot be solved anymore using exact methods. To overcome these problems, heuristic optimization methods can be used. Binitha and Sathya [2012] presented the GA algorithm to solve the convergence problem for local minima or maxima. Further, GA was unable to effectively solve constrained optimization problems.

5. Modified ABC algorithm

5.1 The original ABC

In 2005, Karaboga [2005] developed the ABC algorithm [Basturk and Karaboga, 2006], which is a method for solving optimization problems. The algorithm imitates the behaviors of bees when searching for food (Honeybees). The bee colony population is divided into three groups: employed bees, onlooker bees, and scouts. The employed bees search for food and then came back to share information about food sources to the onlooker bees, which search their nests, inspect the selected food sources, and compare them with the food sources nearby. This algorithm improves the ability of exploring and finding the best solution (optimization algorithm) using scout bees, which guide the mutation process to the algorithm by searching for new food sources in the previously unexplored areas of the survey area to increase the chances of finding better food sources. Then, the selected food sources are discarded and transformed from the employee bees to the scout bees so as to find new food sources. The location of a food source is the value of a possible answer. The number of employed bees and scout bees combined is the total number of possible solutions by finding food sources.

The ABC algorithm is an efficient approach for solving the vehicle routing problem. Nevertheless, this algorithm has not been confirmed as a global solution, but it provides optimal solutions for NP-hard problems [Karaboga and Akay, 2009].

Step 1: Initial population creation of all bees with location of food sources chosen by random selection (initial phase) based on Equation (11) where $i = 1,\ldots SN$, SN is indicate the number of food sources, and $j = 1, \ldots, D$, and D is the dimension of the problem.

$$x_{ij} = x_{\min j} + rand\left[0,1\right]\left(x_{\max j} - x_{\min j}\right) \tag{11}$$

Step 2: Employed bee phase. The employed bees search for new food sources based on Eq. (12), where v_{ij} is the new solution in the next generation, Ø is the value obtained from the randomness in the range of 0-1, $i \in \{1, 2,\ldots, SN\}$, SN is the size of colony, $j \in \{1, 2,\ldots, D\}$, and D is the dimension of the problem. Then, the suitability is calculated based on Eq. (13) if the new position value is better than before to update the position to a new value.

$$v_{ij} = x_{ij} + \varnothing \cdot x_{ij} - x_{kj} \tag{12}$$

$$fit_{ij}\left(\bar{x}\right) = \begin{cases} \dfrac{1}{1+f_i\left(\vec{x}_i\right)}, & if\ f_i \geq 0, \\ 1+\left|f_i\left(\vec{x}_i\right)\right|, & if\ f_i < 0 \end{cases} \tag{13}$$

where x denotes the vector answers from random,
i is the population,
j is the parameter value,
$f_i\,(x_i)$ is the x objective function value,
$fit_{ij}\,(x_i)$ is the fitness value solution of the food source.

Step 3: Onlooker bee phase. The onlooker bees consider the obtained food sources from the employed bees using a probability that can be obtained from equation (14), where p_i is the selection probability of the current solution. If the food source has high probability, it is very likely to be chosen. Then, the onlooker bees send the selected data for calculation in order to find more suitable food sources, just as the employed bees.

$$P_i = \frac{fit_i(x_i)}{\sum_{i=1}^{\tau} fit_i(x_i)} \tag{14}$$

where fit_i is the fitness value of solution i, which is proportional to the nectar amount of the food source in position i, and τ is the number of food sources, which is equal to the number of employed bees or onlooker bees.

Step 4: Scout bee phase. When the original food sources of the employed bees are not selected by the onlooker bees, the scout bees calculate the new food sources by randomly replacing those that were not selected.

Step 5: Condition satisfied. The bees stop searching when finding the best food source. Otherwise, they go back to Step 2 again until max iterations.

The ABC is still a qualification of poor exploitation [Gao et al., 2012]. This causes capturing in the optima areas and results in slower convergence rates, thus tackling various problems. In the past, many researchers modified the basic ABC structure. The proposed variable was named LS-ABC (local search based ABC). The performance was tested on more than 12 standard functions [Sharma and Pant, 2013]. Pathak [2017] presented an enhanced ABC algorithm with local search using an incremental approach for the traveling salesman problem.

5.2 Local search

The local search strategy is a kind of constraint propagation. When solving constraint networks, local search strategies are often modified to discard the states that cannot be solutions and to rank the states that are still solution candidates. This idea has been applied to efficiently explore large neighborhoods [Meseguer, Rossi, and Schiex, 2006]. The local search algorithm starts from a candidate solution, and iteratively moves to a neighbor solution. From the original ABC, in the onlooker bees' states, a neighbor solution is only used for replacement when the onlooker bees find the best neighbor solution [Szeto, 2011].

The local baseline search algorithm starts with an arbitrary solution and ends with a local minimum, which cannot be further improved. During these steps, there are several local search ways. For the best improvement, such as greedy selection, the local search replaces the current solution with the most cost-improving solution after searching the entire neighborhood. The local search method can quickly resolve optimal routing [Prosser and Shaw, 1997]. The limitations of the problem variance were also found to be quite high. We found the method to improve the operators in the steepest descent search strategy to avoid the local optima for reaching global optima as with the guided local search (GLS) algorithm [Kibly et al., 1999]. To probably improve the efficiency of the solutions in the search process,

Guided local search (GLS) is an optimization technique which is an intelligent search algorithm that exploits information to guide the local search in avoiding the local optimum [Voudouris, 1997; Voudouris et al., 2010; Kilby et al., 1999; Arnold and Sörensen, 2019]. The GLS solution modifies action from local search by augmented cost function of minimizing the problem objective function with the cost function to a penalty term that was applied by a penalty vector *p*, where pi is the penalty value of feature *i*. The GLS uses local search to minimize objective function by augmented objective function. Therefore, local search is performed via local search (S, p) function, staring from solution S and then returning to a new solution improved by the augmented objective $h(S)$which spread of penalties as follows:

$$h(S) = O(S) + \lambda \sum_{i \in M} p_i l_i(S) C_i \tag{15}$$

where $O(S)$ is the original objective function problem. The C_i is a cost vector of feature *i*. p_i is the penalty parameter and if the feature is not exhibited in the local optimum, then the penalty value is 0, when the local search is tapped, where the penalty parameters are incremented by 1. Penalty is the indicator involving the feature *i*, which is the distance between customer location and other locations, and λ, a parameters to GLS, represents the relative value of penalties to control the information on the search process with respect to the actual solution cost. Arnold and Sörensen [2019] found that $\lambda = 0.1$ works well. A l_i is a Boonlean indicator in the solution feature *i*. The essential effectiveness of GLS is the penalty parameters that are the costliest features in the current solution and are weighted by the number of times a feature has already been penalized. The penalties of the features are initialized to zero and are incremented for the features that maximize the utility formula. After the improvement method when local search settles, the penalty factor was used to penalize the bad features of *i*. If they keep a local search in local optimization, the current solution, which has the most cost, is penalized by weighing the number of times. They choose the features $i \in S$ for which $C_i/(p_i + 1)$ is the largest among the features in S.

5.3 Proposed algorithm

This study proposed states for improving the performance of the ABC algorithm to solve the bike-sharing relocation problem, which is modified in the scout bee phase using local search based on the neighboring operators. The proposed algorithm is called the GLS-ABC algorithm, and it was demonstrated, as shown in Table 2.

We proposed a modified state so that xi is replaced by the best neighbor solution, which is a GLS that starts from a randomly selected complete instantiation and moves to the next instantiation. This idea may prevent the bad regions of the solution search.

The solution starts searching for the route tour that is closest to the depot. Next, searching for stations makes addition of more routes feasible based on the demand of each station and the capacity of the truck's delivery constraints.

The algorithm finds the best route by utilizing the cost function, which is determined from the distance between two stations and the demand of each station, under truck constraint considering the truck's capacity possible load and unload itself.

For the fitness function, is equal to $1/Z(X_i)$, where $Z(X_i)$ is the sum of the route distances of the food source X_i. Thus, the fitness value inverses the total distance value; as such, the minimized total distance affects the fitness value.

The problem was defined by N bike-sharing stations and a symmetric distance matrix, where $D = [d_{ij}]$ gives the distance between any two stations i and j. The goal is to find the minimum total distance for bike-sharing rebalancing based on the truck constraints; the truck starts from depot and visits each station exactly once, and after pickup or drop off, the truck must return to the depot.

6. Experimental

6.1 Data

The experiments revealed that each depot provided service using capacitated vehicles for relocation bikes. Each truck can carry a maximum of 20 bikes. The illustrated dataset was used; each instance consisted of the coordinate of location, pickup demand, and drop-off demand. The coordinates of the bike-sharing stations were randomly generated in a Euclidean plan. Also, the drop off and pick up demands of the bikes at each station were randomly generated to vary a dataset consisting of

Table 2: Pseudocode of the proposed algorithm

Pseudocode of GLS-ABC
Begin:
Initialization of generate a set of food sources x_i, $i = 1,\ldots, SN$ according to Eq.11
Evaluate each x_i, $i = 1,\ldots, SN$ according to Eq.11
Set $v = 0$ and $l_i = 0$, $i = 1, 2,\ldots, Nb$
While (v < MaxInteration) do
For $i = 1$ to SN do (employed bees)
Select a random solution according to Eq.12
Calculate its fitness value of new food source according to Eq.13
Apply random neighborhood operator, GLS selection by
Penalize the worst distance (i, j) by incrementing $p(i)$
Apply local search using Eq.15
End for
Calculate the probability each food source according to Eq.14
For $i = 1$: (onlooker bees)
select a food source x_i using fitness-based roulette wheel selection method.
Apply neighborhood operator on $x_i \rightarrow \hat{x}$, GLS selection by
Penalize the worst distance (i, j) by incrementing $p(i)$
Apply local search using Eq.15
if $f_i(x_i) < f_i(\hat{x}_i)$,
select $\hat{x}_j$ that is set of food sources with j is maximize of set of food sources,
replace $\hat{x}_j$ with $\hat{x}$,
End if
End for
if l_i is limit, (determine abandoned solution for the scout)
then using a neighborhood operator on $x_i \rightarrow \hat{x}$, and replace x_i with $\hat{x}$.
End if
Memorize the best solution.
$V = V + 1$
End while

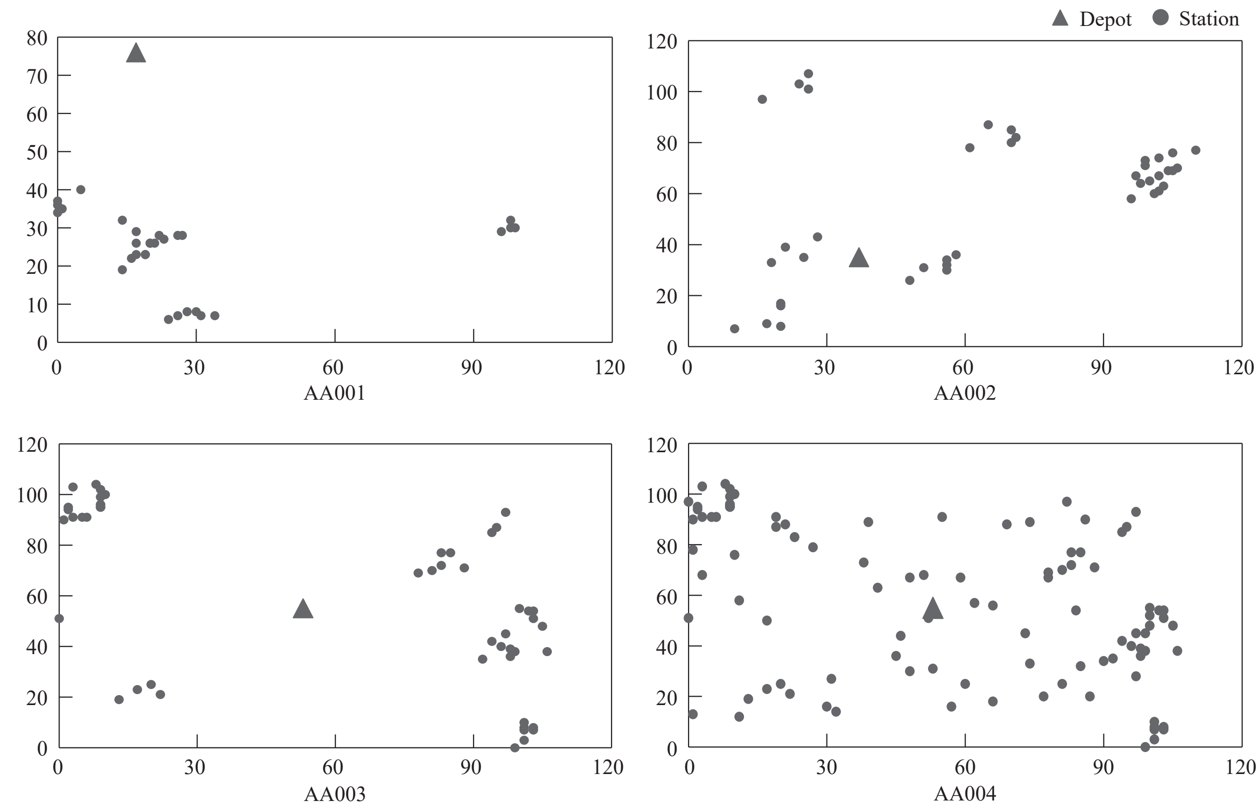

Figure 1: Bike-sharing station distribution at each instance

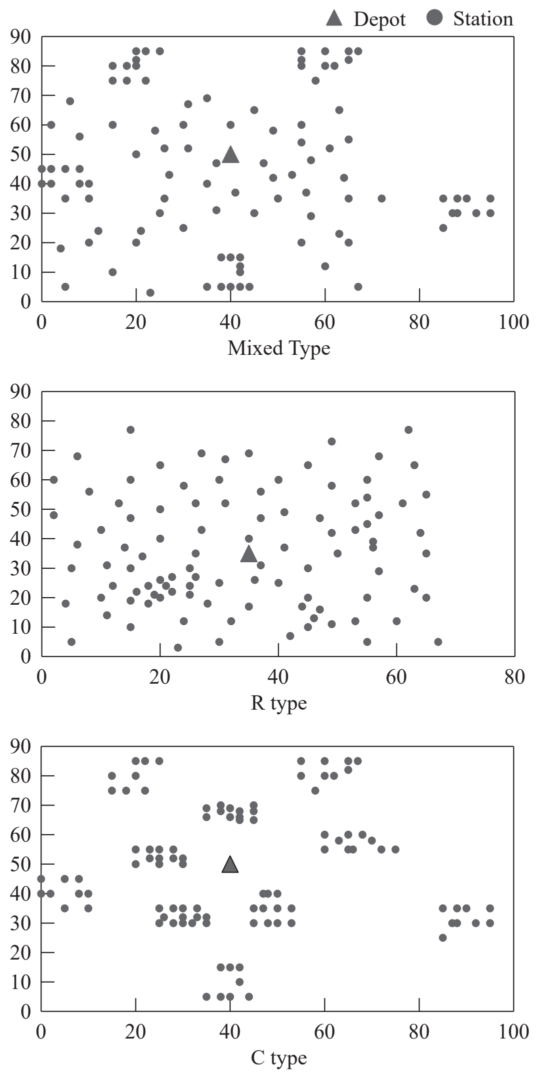

Figure 2: Bike-sharing station distribution at each instance using difference types (Solomon Problem)

four instances as AA001-AA004.

We evaluated the performance of the proposed algorithm based on the locations of the customers using the Solomon problem [Solomon, 2005], which is classified into three types. First, the R data set having randomly distributed bike stations locations was generated in a problem consisting of R101 and R102. Second, the C data set have clustered bike stations location distribution in a problem consisting of C101 and C102. Last, the RC data set have a mix of randomly located and clustered bike stations distribution structures in a problem consisting of RC101, and RC201, where each problem has 100 bike stations. The drop off and pick up demands of the bikes at each station were randomly generated.

The experiments considered how to operate a minimized total distance of the truck routing for bike-sharing relocation.

6.2 Evaluating the performance of the modified algorithm.

We evaluated the performances of the original ABC algorithm (LS-ABC) and proposed ABC algorithm (GLS-ABC). The experimental design of each condition contains 20 replicates (run 20 times) to improve the accuracy and reduce experimental errors. We set the number of employed bees and number of onlookers to be equal to the number of food sources ($\tau = 25$) based on [Koraboga and Basturk, 2008], and each time the algorithm was run for 2,000 iterations [Szeto, Wub, and Ho, 2011]. The experiments were conducted in randomization, and the collected results consisted of the best total distance and average total distance. Then, the results were confirmed by statistically comparing the performances of the difference approaches. The paired *t*-test and analysis of variance method were used. The simple cost operation categories were divided into two groups: fixed cost groups may include the vehicle leasing cost and the diver's salary, and the variable cost group, which is the fuel per meter, and vehicle maintenance. For reduce cost that change directly as the fuel cost per kilometer. Thus, this paper only presented the total travel distance. This paper has raised the issue of improving the routing efficiency for the bike-sharing relocation problem. Therefore, the critical point of the change was compared, namely the cost of fuel per kilometer.

7. Results and discussion

The model for finding out the best solution was coded using Python and run on a computer (Intel core i7 CPU3.80 GHz PC with 16 GB RAM, Windows 10). The optimal solution results of the experiments consisted of the best objective value of the minimum total distance and the average total distance. We measured the performance using two methods: number of stations and data set type. The results of the various data of the number of stations were shown in Table 3, the CPU times were shown in Table 5, and the percentage improvement of the objective value (total distance) was compared between the original ABC and LS- ABC, and the original ABC and GLS- ABC. Furthermore, the *p*-value of the *t*-test of the comparative mean objective values between the algorithms based on the original

Table 3: Comparing ABC, LS-ABC, and GLS-ABC

Instance	Number of stations	Average			Best		
		ABC	LS-ABC	GLS-ABC	ABC	LS-ABC	GLS-ABC
AA01	30	423.9	379.08	361.82	371.91	311.24	301.12
AA02	40	568.22	504.36	500.75	561.86	481.94	460.04
AA03	50	876.67	751.44	728.82	811.62	730.41	721.8
AA04	100	1907.87	1559.15	1530.12	1772.31	1469.9	1447.74

Table 4: Comparison of the improvements of the experiment results for LS-ABC and GLS-ABC

Instance	Average				Best			
	LS-ABC		GLS-ABC		LS-ABC		GLS-ABC	
	%	*P*-Value	%	*P*-Value	%	*P*-Value	%	*P*-Value
AA01	10.57	0.00000000	19.02	0.000	16.3	0.01090	14.65	0.0101000
AA02	3.87	0.00000000	18.12	0.000	14.2	0.00690	11.87	0.0005000
AA03	14.29	0.00000000	10.20	0.000	10.0	0.00000	17.67	0.0000000
AA04	18.28	0.00000000	18.31	0.000	17.1	0.00000	19.80	0.0000000

Table 5: Comparison of the CPU times of the experiment results for ABC, LS-ABC, and GLS - ABC

Instance	CPU time (in seconds)			*P*-Value
	ABC	LS-ABC	GLS-ABC	
AA01	20.56	26.73	57.75	2.30E-11
AA02	20.67	28.13	64.31	3.90E-18
AA03	17.84	27.75	80.45	2.44E-10
AA04	30.70	47.02	120.14	8.19E-19

Table 6: The experiment results of the different types of data sets for ABC, LS-ABC, and GLS-ABC

Instance	Average			Best			*P*-Value
	ABC	LS-ABC	GLS-ABC	ABC	LS-ABC	GLS-ABC	
C101	910.84	882.80	862.80	854.50	822.42	818.70	0.0154093746180301
C202	1038.19	1012.44	1002.44	987.82	947.32	930.10	6.1318625805E-13
R101	1061.59	1023.04	1010.12	970.30	956.94	944.35	0.00276636
RC101	1174.39	1132.31	1110.71	1045.79	1027.40	980.50	0.0168624951943672

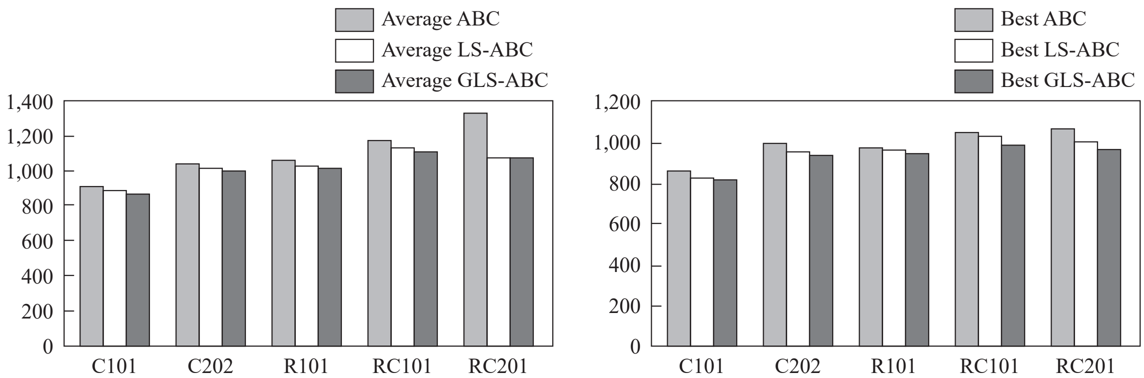

Figure 3: Comparisons of the different types of data sets of the experiment results

ABC was shown in Table 4. From Tables 3 and 6, regarding the test results of the total distance and average tour distance of the bike-sharing relocation problem, the GLS-ABC algorithm improved the total distance by more than 3 % at each instance and on average distance by more than 10 % at each instance. The original ABC was statistically significant with the LS-ABC and the GLS-ABC via p-value < 0.05 at each instance for both total and average distance. Nevertheless, as shown in Table 5, the GLS-ABC took more time than LS-ABC and the original ABC. Overall, the test results show that the GLS-ABC algorithm can produce much better solutions than the original ABC and the LS-ABC in regard to solving bike-sharing relocation problems.

Table 4 shows that the GLS-ABC algorithm improved both the average and best total distances compared with the basic ABC algorithm and LS-ABC algorithm. The essential operation cost is a variable cost of the fuel cost per kilometer. The results showed that the GLS-ABC algorithm is most likely better in reducing the bike-sharing relocation cost.

8. Conclusion

The bike-sharing relocation problem involves customer satisfaction and the benefits of bike-sharing service providers. Also, bike-sharing services are alternative transportation options for many travelers and locals, and they offer an environmentally friendly and healthy transportation method. In this paper, an alternative algorithm was presented to solve the bike-sharing relocation problem. The ABC algorithm could effectively find a solution for the routing problem. We modified the original artificial bee colony to improve the effective solution, and the results showed that the modified algorithm is better than the original ABC algorithm and GLS-ABC in regard to diminishing the bike-sharing relocation problem. The GLS-ABC algorithm could also offer better solutions than those of the original one. The operational costs could be reduced by reducing the vehicle fuel cost. However, the proposed algorithm took a longer time than the original one. In the future, we aim to use the proposed algorithm to solve other real data set in real situations. Also, we aim to breakeven the number of times for rebalancing a day and to develop a different algorithm. Then, one may consider being served with uncertain demand with unknown distributions, which is similar to the real situation.

References

Arnold, F. and Sörensen, K. (2019). Knowledge-guided local search for the Vehicle Routing Problem. *Computers & Operations Research*, Vol. 105, 32-46.

Basturk, B. and Karaboga, D. (2006). An artificial bee colony (ABC) algorithm for numeric function optimization. *Proceedings of the IEEE Swarm Intelligence Symposium 2006*.

Battarra, M., Cordeau, J.-F., and Iori, M., (2014). Pickup-and-delivery problems for goods transportation. *Vehicle routing: problems, methods, and applications*, 2nd ed, Society for Industrial and Applied Mathematics, 161-191.

Cruz, F., Subramanian, A., Bruck, B. P., and Iori, M. (2017). A heuristic algorithm for a single vehicle static bike sharing rebalancing problem. *Computer & Operation Research*, Vol. 79, 19-33.

Erdoğan, G., Battarra, M., and Calvo, R. W. (2015). An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research*, Vol. 245, No. 3, 667-679.

Gajpal, Y. and Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computer & Operation Research*. Vol. 36, No. 12, 3215-3223.

Gao, W., Liu, S., and Huang, L. (2012). A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, Vol. 236, No. 11, 2741-2753.

Japan Tourism Statistics (2020). About trends in visitor arrivals to Japan (Retrieved February 11, 2020 from https://statistics.jnto.go.jp/en/graph/#graph--inbound--travelers--transition).

Katoch, S., Chauhan, S.S. and Kumar, V. (2020) A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 2020, 1-36.

Karaboga, D. (2005). An idea based on honeybee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga, D., Akay, B., (2009). A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation, Vol 214, No. 1, 108-132.

Kilby, P., Prosser, P., and Shaw, P., (1999). Guided local search for the vehicle routing problem with time windows. In Voß, S., Martello, S., Osman, I. H., and Roucairol, C. (eds.) *Meta-Heuristics*, Springer.

Koraboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for mumerical function optimization: artificial bee colony (ABC) alrotithm. *Journal of Global Optimization*, Vol. 39, No. 3, 459-471.

Koraboga, D. and Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, Vol. 8, No. 1, 687-697.

Kumar, A., Kumar, D., and Jarial, S. K. (2017). A review on artificial bee colony algorithms and their applications to data clustering. *Cybernetics and Information Technologies*, Vol. 17, No. 3, 3-28.

Meseguer, P., Rossi, F., and Schiex, T. (2006). Soft constraints. Foundations of Artificial Intelligence, 281-328.

Okoro, E. E., Agwu, O. E., Olatunji, D., and David Orodu, O. D. (2019). Artificial bee colony ABC a potential for optimizing well placement: A review. *Proceedings of the SPE Nigeria Annual International Conference and Exhibition*.

Pathak, N., Mishra, M., and Kushwah, S. P. S. (2017). Improved local search based modified ABC algorithm for TSP problem. *Proceedings of 2017 4th International Conference on Electronics and Communication Systems*. 173-178.

Prosser, P. and Shaw, P. (1997). Study of greedy search with multiple improvement heuristics for vehicle routing problems. *Technical Report RR/96/201*, Department of Computer Science, University of Strathclyde.

Rothlauf, F. (2011). Optimization methods. In *Design of Modern Heuristics*. Natural Computing Series. Springer.

Singla, A., Santoni, M., Bartok, G., Mukerji, P., Meenen, M., and Krause, A. (2015). Incentivizing users for balancing bike sharing systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 723-729.

Sharma, T. K. and Pant, M. (2013). Enhancing the food locations in an artificial bee colony algorithm. *Soft Computing*, Vol. 17, No. 10, 1939-1965.

Shui, C. S. and Szeto, W. Y. (2017). Dynamic green bike repositioning problem: A hybrid rolling horizon artificial bee colony algorithm approach. *Transportation Research Part D: Transport and Environment*, Vol. 60, 119-136.

Solomon, M. M. (2005). Best known solutions identified by heuristics, Northeastern University, Massachusetts, Boston.

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

Usama, M., Zahoor, O., Shen, Y., and Bao, Q. (2020). Dockless bike-sharing system: Solving the problem of faulty bikes with simultaneous rebalancing operation. *Journal of Transport and Land Use*, Vol. 13, No. 1, 491-515.

Voudouris, C. (1997). Guided local search for combinatorial optimisation problems, Ph.D. dissertation, Department of Computer Science, University of Essex, UK.

Voudouris, C., Tsang, E. P., and Alsheddy, A. (2010). *Guided local search*, *Handbook of Metaheuristics*. Springer US.

Wassan, N. A. and Nagy, G. (2014). Vehicle routing problem with deliveries and pickups: Modeling issues and metaheuristics solution approaches. *International Journal of Transportation*, Vol. 2, No. 1, 95-110.

(Received February 11, 2021; accepted March 17, 2021)