Songklanakarin J. Sci. Technol.
41 (1), 151-158, Jan. – Feb. 2019

http://www.sjst.psu.ac.th

Original Article

An artificial bee colony algorithm for the vehicle routing problem with backhauls and time windows

Tanawat Worawattawechai[1*], Boonyarit Intiyot[1], and Chawalit Jeenanunta[2]

[1] *Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Pathum Wan, Bangkok, 10330 Thailand*

[2] *School of Management Technology, Sirindhorn International Institute of Technology, Thammasat University, Khlong Luang, Pathum Thani, 12120 Thailand*

Received: 16 August 2016; Revised: 11 August 2017; Accepted: 8 October 2017

Abstract

The vehicle routing problem with backhauls and time windows (VRPBTW) aims to find a feasible vehicle route that minimizes the total traveling distance while imposing capacity, backhaul, and time-window constraints. We present an enhanced artificial bee colony algorithm (EABCA), which is a meta-heuristic, to solve this problem. Three strategies - a forbidden list, the sequential search for onlookers, and the combination of 1-move intra-route exchange and λ-interchange technique - are introduced for EABCA. The proposed method was tested on a set of benchmark instances. The computational results show that the EABCA can produce better solutions than the basic ABCA, and it discovered many new best-known solutions.

Keywords: meta-heuristic, artificial bee colony, backhaul, time window, vehicle routing problems

1. Introduction

The vehicle routing problem with backhaul and time window (VRPBTW) is extended from the vehicle routing problem with backhaul (VRPB) by adding a specified service time window for each customer. There are three main constraint categories for VRPBTW model, namely capacity constraints, backhaul constraints and time window constraints. For the capacity constraints, the number of customers serviced by a vehicle is restricted by the capacity of the vehicle. For the backhaul constraints, the vehicles serve all demands of the linehual customers and the same vehicles also pick up demands from the backhaul customers. In addition, the backhaul customers cannot be served before linehaul customers. For the time window constraints, the vehicle arrival time at each customer must not exceed the upper bound of the customer's time window. In general, the VRPBTW is a class of the NP-hard combinatorial optimization problems, which is too difficult to solve exactly within a reasonable time. Consequently, there are many heuristic methods proposed to get a near optimal solution for this problem.

An increasing number of the publications on heuristic approaches for vehicle routing problem have been developed for the past two decades. However, only few studies have been devoted to the VRPBTW. A brief review of these studies is divided into two parts based on the types of the proposed methods, namely non-meta-heuristic methods and meta-heuristic methods.

A few non-meta-heuristic methods were proposed to solve VRPBTW. Thangiah *et al.* (1996) presented a push forward insert heuristic (PFIH). This algorithm applied an insertion heuristic for route construction and improved solution by λ-interchange and 2-opt* exchange procedures to solve VRPBTW problems. The algorithm was tested on benchmark instances of Gélinas *et al.* (1995). Although the solutions of PFIH were within 2.5% of the optimum on average, PFIH almost always gave worse results than average for large-sized problems. Ropke and Pisinger (2006) transformed the VRPBTW into the VRPB by ordering the routes

*Corresponding author
Email address: i.am.t.tanawat@gmail.com

according to time window constraints and then used a neighbor search algorithm to solve the problem. Although the unified heuristic could improve the best-known solution for many instances and decrease the necessary number of vehicles, the computational time increases considerably with the problem size. Worawattawechai *et al.* (2016a) presented a nearest urgent candidate heuristic (NUC) for the VRPBTW. This algorithm starts by sorting the customers according to the urgency of delivery before adding into the route by considering their closeness while a candidate list technique is used to enforce the urgency order. Although NUC heuristic performed better than the PFIH, it underperformed the existing algorithm for the large-sized problems. In general, non-meta-heuristic methods can achieve good optimization results quickly while it is relatively simple and easy to apply to the problems. However, the efficacy of these methods decreases when the problem size becomes large.

There are more studies that focused on meta-heuristic methods for VRPBTW. Potvin *et al.* (1996b) described a genetic algorithm (GA) coupled with a greedy insertion heuristic to find a good insertion order of the customers. The results showed that GA solutions were within 1% of the optimum on average by using benchmark instances of Gélinas *et al.* (1995). However, it was very expensive in terms of computational resources, and it was frequently prone to premature converging. Cho and Wang (2005) presented a meta-heuristic which is based on an acceptable threshold combined with modified nearest neighbor and exchange procedures for solving the VRPBTW. Although this method could decrease the computational time, it underperformed compared with the GA proposed by Potvin *et al.* (1996b). Küçükoğlu and Öztürk (2014) introduced a differential evolution algorithm (DEA) for VRPBTW and applied it for a catering firm.DEA was tested with several benchmark problems. The results showed that this algorithm could obtain some new best-known solutions. Since DEA was originally designed for continuous problems, it is hard to find a good encoding procedure to adapt DEA to integer problems like VRPBTW. Later, Küçükoğlu and Öztürk (2015) proposed an advanced hybrid meta-heuristic algorithm (HMA), which combined tabu search algorithm and simulated annealing algorithm to obtain more effective solutions for the VRPBTW. The results indicated that the HMA performed better than the DEA. However, one of the disadvantages of the hybrid algorithm was that it took a lot of computational time. Worawattawechai *et al.* (2016b) proposed the nearest neighbor with roulette wheel selection method (NNRW) as an initial solution algorithm for the cuckoo search (CS) algorithm. The result reported that the CS algorithm could produce better solutions than the best-known solutions for the majority of small- and medium-sized instances. However, it did not perform as well for large problems.

Artificial bee colony algorithm (ABCA) is another meta-heauristic method that has been applied to VRP. It was first introduced by Karaboga (2005). It is firstly applied to the capacitated vehicle routing problems (CVRP) by Szeto *et al.* (2011) with some enhancements. The results show that the enhanced version of ABCA outperformed the original one, and it could produce good solutions when compared with the existing heuristics. Alzaqebah *et al.* (2016) presented the modified artificial bee colony for the vehicle routing problems with time windows (VRPTW). In this study, the list of abandoned solutions was used to generate new solutions. The results showed that the modified ABCA obtained good results when compared with the best-known results. An improved artificial bee colony algorithm for a real case in Dalian was introduced by Yu *et al.* (2016). In this version of ABCA, three strategies were applied, namely an adaptive strategy, a crossover operation, and a mutation operation. The results showed that some of solutions were better than the best-known solution when tested on benchmark problems of Solomon (1987) for VRPTW.

There are many reasons that motivate the authors to use ABCA to solve VRPBTW in this paper. Firstly, ABCA was successfully applied to VRP and VRPTW as described in the above paragraph. Secondly, ABCA is a meta-heuristic, which means the exploring area of the solution space is larger than non-meta-heuristics (PFIH, unified heuristic, NUC heuristic). Thus, it can achieve good optimization results, especially in the large-sized problem. Thirdly, ABCA is a population-based heuristic which starts with a number of initial solutions. Therefore, it can explore more in the solution space and get more chance to obtain the better solutions than a non-population-based heuristic (e.g. HMA). Moreover, a population-based heuristic can be enhanced with parallel computing or distributed computing. Finally, ABCA can prevent the search from premature convergence problem which is the weakness of other population-based heuristics (e.g. GA and DEA). This is because, in the scout bee stage, the stalled solutions are removed from the population and a new random - generated solution is added to the population. This process also amplifies global search capability.

There are a few studies (Tuntitippawan & Asawa rungsaengkul 2016a, 2016b) that apply ABCA for solving VRPBTW. Tuntitippawan and Asawarungsaengkul (2016a) applied ABCA to small and medium problems and Tuntitip pawan and Asawarungsaengkul (2016b) applied ABCA to small, medium, and large problems. However, the computational results showed that it still underperformed compared with the existing heuristics in many instances, especially in the large-scale problems. It is necessary to extend the exploration on the solution space or, equivalently, to expand the capability of the neighborhood search. Therefore, we introduce the enhanced artificial bee colony algorithm (EABCA) by applying a forbidden list strategy to prevent duplicated initial solutions (which initially extends the exploration on the solution space), the sequential search strategy for onlookers to explore the neighborhood near the high-quality food source, and the intra-route and inter-route exchange combination strategy to obtain the better solutions. Moreover, the parametrization is studied in this paper.

2. Enhanced Artificial Bee Colony Algorithm for VRPBTW

2.1 The general concept of an artificial bee colony

The artificial bee colony is inspired by the intelligent finding food sources behavior of the honey bees around the hives proposed by Karaboga (2005). A colony of the bees consists of three types of bees: employed bees, onlookers and scouts. The employed bees search for available nectar sources and share this information with the onlookers via a waggle dance at the dancing area. The onlookers select the food sources by evaluating quality of nectar sources from the waggle dance to be further explored. When the quality of food sources is not improved within a time limit, the employed bees abandon the food source and turn into scout bees to find new food sources.

The ABCA starts by generating a number of nectar sources (initial solutions) and assigning an employed bee to each food source. Each employed bee explores new food source near its original food source (neighborhood search) and measures the nectar amounts (fitness value). If the new source has more nectar, it will replace the old one. Then the employed bees return to the hive with the information of the updated food sources, which is shared with the onlookers by the waggle dance. Each onlooker selects a food source with a probability that depends on the nectar amounts (the roulette wheel method). In particular, a food source with higher nectar amounts has a higher probability to be selected by an onlooker than ones with lower nectar amounts. Then each onlooker finds a new food source around the selected food sources (neighborhood search) and evaluates the amount of nectar. The employed bee will abandon its old food source and go to the new one if it has more nectar. In the case that the quality of food source is not improved within a time limit, the employed bee will also abandon the old food source and become a scout bee that searches for the new food source by randomly generating a new solution. After the scout bee finds a new food source, it becomes an employed bee again. This process will repeat until a stopping criterion is reached.

2.2 Main steps

The steps of the EABCA for solving the VRPBTW model can be described as follows:

Step 1 Generate a set of initial solutions (food sources) by the nearest neighbor with roulette wheel selection method. The forbidden list strategy is also applied in this process. (Details in Section 2.3) Then assign each food source to each employed bee.

Step 2 Evaluate the fitness of each solution and record the global best solution.

Step 3 Apply the neighborhood search on each food source. An employed bee abandons its old food source if a new one with better fitness is found. Otherwise, increment the time limit counter of the food source.

Step 4 For each onlooker, select a food source by using the fitness-based roulette wheel selection method and improve the food source by the neighborhood search. If the onlooker bee finds a new neighbor solution with better fitness, the employed bee associated with the food source abandons its old food source and goes to the new one.

Step 5 Update the global best solution if a solution has better fitness than the current best one.

Step 6 Check the time limit counter of each food source. If it reaches the predetermined number, the food source is replaced by a new randomly generated solution.

Step 7 If the number of iterations reaches the maximum, then the algorithm finishes. Otherwise, go back to Step 3.

2.3 Initial solution construction

Küçükoğlu and Öztürk (2015) proposed an improved nearest neighbor heuristic for constructing an initial solution for VRPBTW. They computed the closeness of customer i to customer j by using $proximity_{ij}$, which is defined as: $proximity_{ij} = \alpha c_{ij} + \beta h_{ij} + \gamma v_{ij}$, where α, β, γ denote weight parameters such that $\alpha + \beta + \gamma = 1$, $\alpha \geq 0$, $\beta \geq 0$, $\gamma \geq 0$; c_{ij} denotes the traveling time from customer i to customer j; h_{ij} denotes the idle time before servicing customer j after customer i; and v_{ij} denotes the urgency of delivery to customer j after customer i expressed as the time remaining until the vehicle's last possible service start for customer j. Then the closeness of customers i and j, denoted by $closeness_{ij}$, is defined as the reciprocal of $proximity_{ij}$.

This paper adopts the above definition of closeness and uses it in the construction of the initial solutions. Each initial solution is constructed by the nearest neighbor with roulette wheel selection method proposed by Worawatta wechai *et al.* (2016b). An initial solution construction always starts a tour with the depot, and then finds the next customer by spinning the roulette wheel. If the next customer violates the constraints, we spin the roulette wheel again to find a new one. If we cannot find the next customer without violating constraints, we end this tour and begin a new tour. This process is repeated until all customers are served.

Using the roulette wheel method alone can cause duplicate initial solutions, which hinders the exploration space down the road. In EABCA, the forbidden list strategy is applied to prevent this problem. Initially, the forbidden list is empty. Subsequently, after a new feasible initial solution is obtained, the solution will be checked with the forbidden list. If the solution is not in the list, then it is added to the forbidden list. Otherwise, the solution will be abandoned. The process is repeated until the number of solutions in the forbidden list reaches the specified number of initial solutions.

2.4 Fitness function

In this paper, we compute the Euclidean distance between customer i and customer j using the following formula (Kohl *et al.*, 1997, 1999):

$$c_{ij} = \frac{\left\lfloor 10\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)\right\rfloor}{10}$$

where (x_i, y_i) is the coordinate of customer i and (x_i, y_i) is the coordinate of customer j. The traveling time between two customers is assumed to be the same as the distance between them.

The fitness value (f) of a solution is a reciprocal of the total distance traveled by all vehicles in the solution.

2.5 Neighborhood search

The local search in the ABCA proposed by Tuntitippawan and Asawarungsaengkul (2016b) only uses the λ-interchange, which is an inter-route operator that considers two routes at once. To extend the search ability, the EABCA can either randomly apply λ-interchange or 1-move intra-route exchange, which work on a single route, for its neighborhood search. Since the 1-move operator improves the solution by deleting a customer and then inserting it into the same route, it helps rearranging the customer in the route. The experimental parameter testing discussed in Section 3.1 indicates that this setting gives better solution than using λ-interchange alone (Figure 5). The details of both types of operator are explained below.

The 1-move intra-route exchange is a practical operator for the traveling salesman problem. We adapt this operator to improve each route in the vehicle routing solution by removing one customer from a route and insert it back to the same route in a different position. An example of the 1-move intra-route exchange is given in Figure 1.

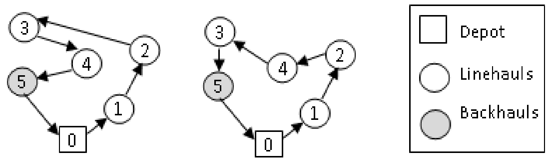

Figure 1. An example of the 1-move intra-route exchange.

λ-interchange is a generalization of a relocation operator for the vehicle routing problem proposed by Osman (1993). The idea is to exchange a subset of customers of size x with a subset of customers of size y from a different route, which can be represented by the operator (x, y) where x and y are nonnegative integers not bigger than λ. In this paper, we use λ = 4. For examples, one possible operator is (1, 0) which means moving one customer in the one route to another route as shown in Figure 2. Another possible operator is (2, 1) which means exchanging two customers in the one route with one customer in another route as shown in Figure 3.

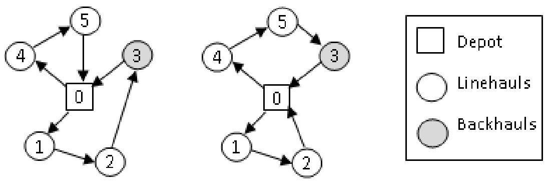

Figure 2. An example of the operation (1, 0).

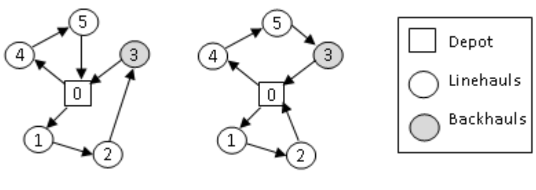

Figure 3. An example of the operation (2, 1).

2.6 Fitness-based roulette wheel selection method

The fitness-based roulette wheel selection method is applied in the food source selection process. In this method, each onlooker selects a food source to explore according to the probability that depends on the nectar amount (or fitness value). After selecting a food source, each onlooker finds a new food source around the selected food source and evaluates the amount of nectar.

In the original version, if there are many onlooker bees selecting the same food source, each onlooker individually searches for a new food source and the old food source is replaced by the best of those new food sources. In our enhanced version, if there are many onlooker bees selecting the same food source, they will be queued up for searching a new food source. The search can only be performed by one onlooker bee at the time. If the previous onlooker bee finds a new better food source, the next onlooker bee will start from the newly found food source and look for a better one. Otherwise, the next onlooker bee will start from the same food source as the previous one. In this way, algorithm will be given opportunities to be further explored in good regions of the solution.

3. Data Tests and Computational Results

The EABCA was programmed in Microsoft Visual C# 2010 Express and executed on a 2.4 GHz Intel i7 Duo with 8 GB memory. We tested the algorithm on a set of benchmark instances developed by Gélinas *et al.* (1995) for VRPBTW.

3.1 Parameter setting

A small study on parametrization of our algorithm is carried out and shown in this section. The crucial parameters $(\alpha,\ \beta,\ \gamma,\ \lambda)$ are varied and their solutions are compared using a randomly selected large problem with 10% backhauls.

The parameters α, β, and γ are the proportion weights for traveling time, idle time, and urgency of delivery respectively when $\alpha+\beta+\gamma=1,\ \ \alpha\geq 0, \beta\geq 0, \gamma\geq 0.$ Therefore, we analyzed the ratio of these parameters instead of individual value analysis. The relationship of the fitness and some ratios of α, β, γ parameters is shown in Figure 4. The experiment indicated that the performance of this algorithm is better when α parameter is weighted more than the others, and it can produce the best solution when $\alpha : \beta : \gamma = 0.4 : 0.3 : 0.3$. Thus, these parameters are set as $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$ in this paper.

The relationship between the fitness value and parameter λ is shown in Figure 5. The smaller λ is, the more difficult it is for EABCA to obtain better solution since the number of customers to be exchanged between routes is limited. Thus, the value of parameter λ = 4 is set in this paper. Moreover, the comparison of λ-interchange with and without 1-move intra-route is also shown in this figure. The experiment indicated that the λ-interchange with 1-move intra-route can produce better solution when compared with the λ-interchange without 1-move intra-route. Thus, the 1-move intra-route can help improve the algorithm performance.

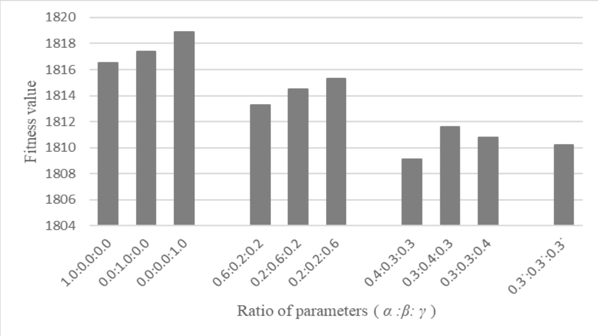

Figure 4. The relationship between the fitness value and the ratio of parameters α, β, and γ.

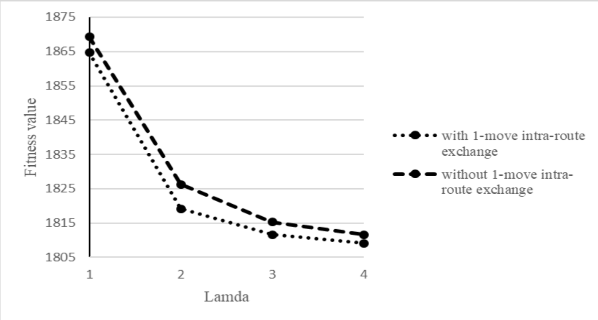

Figure 5. The relationship between the fitness value and parameter λ, and comparison λ-interchange between with and without 1-move intra-route.

Both number of employed bees and the number of onlooker bees are set to be 50, which is recommended by Karaboga and Basturk (2008) for good performance of ABC. For the other parameters, we set them as follows: $\textit{the limit time} = 20$, and maximum number of iterations = 200.

3.2 Computational results

Tables 1-3 compare the results of EABCA with the original ABCA (Tuntitippawan & Asawarungsaengkul, 2016b), HMA (Küçükoğlu & Öztürk, 2015), and DEA (Küçükoğlu & Öztürk, 2014). In addition, the EABCA solutions are compared with the best-known solutions that are collected from many papers, namely Küçükoğlu and Öztürk (2014), (2015), Potvin *et al.* (1996b), Ropke and Pisinger (2006), Thangiah *et al.* (1996), Tuntitippawan and Asawarung saengkul (2016b), Worawattawechai *et al.* (2016a), (2016b). The NV column represents the number of vehicles used in the solution. The best distance of the proposed algorithm from 10 independent runs is shown in the Best Distance column. The %Gap_BKS column denotes the gap percentage between the considered solution and the best-known solution. A negative number in this column means the considered algorithm obtained a new best-known solution. Specifically, the %Gap_BKS is computed by the formula:

$$\%Gap\_BKS = \frac{(\textit{the considered solution}) - (\textit{the best known solution})}{\textit{the best known solution}} \times 100.$$

Table 1. Computational results of the EABCA in VRPBTW with 25 customers.

Problem	BH. (%)	Optimal Solution	BKS	EABCA		ABCA [e]	HMA [c]	DEA [f]	%Gap_BKS			
			Distance	Best Distance	NV	Distance	Distance	Distance	EABCA	ABCA [e]	HMA [c]	DEA [f]
R101	10	643.4[c]	643.4[a]	643.4	9	643.4	643.4	643.4	0.00%	0.00%	0.00%	0.00%
	30	711.1[c]	717.0[b]	721.8	10	721.8	721.8	721.8	0.67%	0.67%	0.67%	0.67%
	50	674.5[c]	676.8[f]	676.8	10	676.8	676.8	676.8	0.00%	0.00%	0.00%	0.00%
R102	10	563.5[c]	563.5[c]	563.5	7	563.5	563.5	565.3	0.00%	0.00%	0.00%	0.32%
	30	622.3[c]	628.1[c]	628.1	9	628.1	628.1	629.0	0.00%	0.00%	0.00%	0.14%
	50	584.4[c]	584.4[c]	584.4	8	584.4	584.4	585.3	0.00%	0.00%	0.00%	0.15%
R103	10	476.6[c]	476.6[e]	476.6	5	476.6	478.8	489.0	0.00%	0.00%	0.46%	2.13%
	30	507.0[c]	507.0[c]	507.0	7	507.0	507.0	510.9	0.00%	0.00%	0.00%	0.77%
	50	475.6[c]	483.0[c]	483.0	6	483.0	483.0	495.0	0.00%	0.00%	0.00%	2.48%
R104	10	452.5[c]	452.5[d]	452.5	5	453.8	453.8	459.1	0.00%	0.29%	0.29%	1.46%
	30	467.6[c]	468.5[c]	468.5	6	468.5	468.5	469.6	0.00%	0.00%	0.00%	0.23%
	50	446.8[c]	446.8[c]	446.8	5	446.8	446.8	458.7	0.00%	0.00%	0.00%	2.66%
R105	10	565.1[c]	565.1[a]	565.1	7	565.1	565.1	565.1	0.00%	0.00%	0.00%	0.00%
	30	623.5[c]	623.5[c]	623.5	8	628.0	623.5	630.2	0.00%	0.72%	0.00%	1.07%
	50	591.1[c]	591.1[d]	591.1	8	591.1	592.1	598.5	0.00%	0.00%	0.17%	1.25%

[a] Obtained from Potvin *et al.* (1996b)
[b] Obtained from Thangiah *et al.* (1996)
[c] Obtained from Küçükoğlu and Öztürk (2015)
[d] Obtained from Worawattawechai et al (2016a)
[e] Obtained from Tuntitippawan and Asawarungsaengkul (2016)
[f] Obtained from Küçükoğlu and Öztürk (2014)

Table 2. Computational results of the EABCA in VRPBTW with 50 customers.

Problem	BH. (%)	Optimal Solution	BKS	EABCA		ABCA [e]	HMA [c]	DEA [f]	%Gap_BKS			
			Distance	Best Distance	NV	Distance	Distance	Distance	EABCA	ABCA [e]	HMA [c]	DEA [f]
R101	10	1122.3[c]	1133.3[g]	1133.3	15	1134.0	1135.8	1138.3	0.00%	0.06%	0.22%	0.44%
	30	1191.5[c]	1191.6[c]	1191.6	16	1191.6	1191.6	1245.8	0.00%	0.00%	0.00%	4.55%
	50	1168.6[c]	1183.9[a]	1183.9	16	1183.9	1183.9	1183.9	0.00%	0.00%	0.00%	0.00%
R102	10	974.7[c]	976.5[e]	976.5	12	976.5	976.8	978.7	0.00%	0.00%	0.03%	0.23%
	30	1024.8[c]	1024.8[b]	1054.6	14	1054.6	1046.0	1046.0	2.91%	2.91%	2.07%	2.07%
	50	1057.2[c]	1059.7[a]	1059.7	14	1059.7	1061.6	1153.0	0.00%	0.00%	0.18%	8.80%
R103	10	811.4[c]	815.5[c]	812.3	9	821.6	815.5	831.1	-0.39%	0.75%	0.00%	1.91%
	30	882.8[c]	887.1[e]	886.2	11	887.1	889.3	895.1	-0.10%	0.00%	0.25%	0.90%
	50	882.1[c]	885.1[e]	883.0	11	885.1	887.7	887.7	-0.24%	0.00%	0.29%	0.29%
R104	10	-	687.7[c]	685.9	7	-	687.7	688.7	-0.26%	-	0.00%	0.15%
	30	-	736.8[c]	734.8	8	-	736.8	737.7	-0.27%	-	0.00%	0.12%
	50	733.6[c]	734.5[g]	733.6	8	739.3	738.2	742.2	-0.12%	0.65%	0.50%	1.05%
R105	10	970.6[c]	972.8[f]	976.2	11	985.2	978.5	972.8	0.35%	1.27%	0.59%	0.00%
	30	1007.5[c]	1024.7[e]	1019.9	12	1024.7	1026.7	1030.0	-0.47%	0.00%	0.20%	0.52%
	50	993.4[c]	993.4[e]	993.4	11	993.4	996.2	1022.2	0.00%	0.00%	0.28%	2.90%

[a] Obtained from Potvin *et al.* (1996b)
[b] Obtained from Thangiah *et al.* (1996)
[c] Obtained from Küçükoğlu and Öztürk (2015)
[d] Obtained from Worawattawechai *et al.* (2016a)
[e] Obtained from Tuntitippawan and Asawarungsaengkul (2016)
[f] Obtained from Küçükoğlu and Öztürk (2014)
[g] Obtained from Worawattawechai *et al.* (2016b)

Table 3. Computational results of the EABCA in VRPBTW with 100 customers.

Problem	BH. (%)	Optimal Solution	BKS	EABCA		ABCA [e]	HMA [c]	DEA [f]	%Gap_BKS			
			Distance	Best Distance	NV	Distance	Distance	Distance	EABCA	ABCA [e]	HMA [c]	DEA [f]
R101	10	1767.9[c]	1811.6[f]	1809.1	24	1818.6	1811.6	1811.6	-0.14%	0.39%	0.00%	0.00%
	30	1877.6[c]	1885.2[d]	1885.0	24	1904.5	1891.1	1925.9	-0.01%	1.02%	0.31%	2.16%
	50	1895.1[c]	1905.9[a]	1921.2	25	1928.2	1911.2	1930.2	0.80%	1.17%	0.28%	1.27%
R102	10	1600.5[c]	1623.7[c]	1620.8	20	1640.7	1623.7	1649.8	-0.18%	1.05%	0.00%	1.61%
	30	1639.2[c]	1705.6[g]	1693.4	21	1717.3	1724.0	1758.2	-0.72%	0.69%	1.08%	3.08%
	50	1721.3[c]	1746.0[b]	1738.7	21	1752.2	1759.8	1777.1	-0.42%	0.36%	0.79%	1.78%
R103	10	-	1346.9[c]	1355.2	17	-	1346.9	1356.3	0.62%	-	0.00%	0.70%
	30	-	1385.9[c]	1408.6	17	-	1385.9	1389.2	1.64%	-	0.00%	0.24%
	50	-	1456.5[h]	1463.7	18	-	1465.0	1465.0	0.49%	-	0.58%	0.58%
R104	10	-	1084.2[h]	1119.6	13	-	1093.4	1105.4	3.27%	-	0.85%	1.96%
	30	-	1136.6[c]	1148.6	14	-	1136.6	1146.5	1.06%	-	0.00%	0.87%
	50	-	1187.7[d]	1207.4	14	-	1189.6	1199.6	1.66%	-	0.16%	1.00%
R105	10	-	1516.0[c]	1514.3	18	-	1516.0	1527.7	-0.11%	-	0.00%	0.77%
	30	-	1581.5[c]	1594.5	17	-	1581.5	1582.6	0.82%	-	0.00%	0.07%
	50	-	1604.1[c]	1607.2	18	-	1604.1	1608.6	0.19%	-	0.00%	0.28%

[a] Obtained from Potvin *et al.* (1996b)
[b] Obtained from Thangiah *et al.* (1996)
[c] Obtained from Küçükoğlu and Öztürk (2015)
[d] Obtained from Worawattawechai *et al.* (2016a)
[e] Obtained from Tuntitippawan and Asawarungsaengkul (2016)
[f] Obtained from Küçükoğlu and Öztürk (2014)
[g] Obtained from Worawattawechai *et al.* (2016b)
[h] Obtained from Ropke and Pisinger (2006)

Tables 1-3 show the comparisons for small-, medium-, and large-sized problems respectively. The results obtained from the comparison can be summarized as follows.

- When compared with the original ABC, the EABCA obtained 34 equivalent or better solutions out of 34 problems presented in the ABCA paper (100%).

- When compared with the HMA, the EABCA obtained 36 equivalent or better solutions out of 45 problems presented in the HMA paper (80%).
- When compared with the DEA, the EABCA obtained 38 equivalent or better solutions out of 45 problems presented in the DEA paper (84%).

The results show that the EABCA is superior to ABCA in terms of solution quality, and it is competitive with the other heuristics in the literature.

To evaluate the efficiency of EABCA, the comparison between the best-known solutions in the literature and the solutions obtained from the proposed heuristic in this paper is also shown in %Gap_BKS column of Tables 1-3. The results obtained from the comparison can be summarized as follows.

- For small problems with 25 customers in Table 1, the EABCA obtained 14 solutions that were equivalent to the best-known solutions out of 15 instances. In other words, the EABCA solutions for all benchmark instances were equivalent to the best-known solutions except for only one from the R101 problem with 30% backhauls.
- For medium problems with 50 customers in Table 2, the EABCA obtained 13 solutions that were equivalent to or better than the best-known solutions out of 15 instances. Moreover, the proposed algorithm could find 7 new best-known solutions. In general, the EABCA outperformed the existing algorithms in medium problem set.
- For large problems with 100 customers in Table 3, most of the EABCA solutions were not as good as the best-known solutions except for 6 cases where the new best-known solutions were obtained, namely the R101 problem with 10% and 30% backhauls, all problems in the R102, and the R105 problem with 10% backhauls.

Summarily, the EABCA outperformed the existing algorithms in terms of solution quality in many problems as it obtained 33 equivalent or new best-known solutions out of 45 instances (73.33%) while others did not perform as well (ABCA 58.82%, HMA 53.33%, and DEA 13.33%). Moreover, our algorithm found the optimal solutions for some instances. The average computational time of EABCA for 25, 50, and 100 customers are 15.16, 87.64, and 275.47 (seconds) respectively. The gap of the total distance between the best-known solutions and the proposed solutions are within 0.5% of the best-known solutions (0.06% for 25 nodes, 0.12% for 50 nodes and 0.48% for 100 nodes). It is computed by the formula:

$$\%\mathrm{Gap}_{\mathrm{total}} = \frac{(\textit{the summation of all EABC solution}) - (\textit{the summation of all best known solution})}{\textit{the summation of all best known solution}} \times 100.$$

3.3 Result discussion

When comparing the results of enhanced version of ABCA with the original one proposed by Tuntitippawan and Asawarungsaengkul (2016b), the EABCA was superior to the original ABCA in terms of solution quality. We speculated that the forbidden list strategy in generating process, the sequential search strategy for onlooker bees, and the intra-route and inter-route exchange combination strategy for the local search in the EABCA indeed helped extend the exploration on the solution space to obtain the better solutions. Note that although the sequential search of onlookers increases the chance of finding great solutions, it also leads to larger computational time. Further study is needed to analyze the tradeoffs and compare the computational time with the original ABCA.

When comparing the results of EABCA with the other methods in terms of solution quality, we found that the performance of our algorithm is better than the HMA and DEA for small- and medium-sized problems while comparable with the HMA and the DEA in the large-sized problems. We speculated that there are four main reasons EABCA contributes the successful results. First, the EABCA is a population-based heuristic which starts with a number of unduplicated initial solutions. Therefore, it can explore more in the solution space and get more chance to obtain the better solutions. Second, the EABCA applied the combination of intra-route and inter-route exchange as the neighborhood search. Hence, this strategy can extend the regions of the search space to increase the chance for finding a better solution. Third, the high-quality solutions are used more often than the low-quality ones to produce an improved solution in the onlooker bee stage. Thus, the regions of the search space are searched in shorter time and in detail. Fourth, the stalled solutions are removed from the population and a new solution from random generating is added to the population in the scout bee stage. This process provides global search ability and prevents the search from premature convergence problem.

4. Conclusions

In this study, we present the enhanced artificial bee colony algorithm (EABCA) to solve the VRPBTW problem. Three strategies are proposed in EABCA, which are a forbidden list, the sequential search for onlookers, and the combination of 1-move intra-route exchange and λ-interchange technique. The computational results show that EABCA was superior to original ABCA proposed by Tuntitippawan and Asawarungsaengkul (2016b), and it was competitive with the other heuristics in terms of solution quality. Moreover, EABCA solutions were compared with the best-known solutions in the literature. Results show that EABCA obtained 33 equivalent or new best-known solutions out of 45 problems (73.33%). In general, when compared with existing algorithms, EABCA gave better performance in medium-sized problems and comparable performance in small-sized problem. Thus, the EABCA can be applied effectively to small- and medium-sized problems.

References

Alzaqebah, M., Abdullah, S., & Jawarneh, S. (2016). Modified artificial bee colony for the vehicle routing problems with time windows. *Springer Plus*, *5*(1), 1298.

Bäiysy, O., & Gendreau, M. (2002). Tabu search heuristics for the vehicle routing problem with time windows. *Sociedad de Estadistica e Investigación Operativa*. *10*(2), 211-237.

Cho, Y., & Wang, S. (2005). A threshold accepting meta-heuristic for the vehicle routing problem with backhauls and time windows. *Journal of the Eastern Asia Society for Transportation Studies, 6,* 3022–3037.

Gélinas, S., Desrochers, M., Desrosiers, J., & Solomon, M. (1995). A new branching strategy for time constrained routing problems with application to backhauling. *Annals of Operations Research, 61*, 91–109.

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

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8,* 687–697.

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., & Soumis, F. (1999). 2 –Path cuts for the vehicle routing problem with time windows. *Transportation Science*, *33*(1), 101–116.

Kohl, N., & Madsen, O. B. G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations Research*, *45*(3), 395–406.

Küçükoğlu, İ., & Öztürk, N. (2014). A differential evolution approach for the vehicle routing problem with backhauls and time windows. *Journal of Advanced Transportation, 48,* 942–956.

Küçükoğlu, İ., & Öztürk, N. (2015). An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. *Computers and Industrial Engineering*, *86,* 60-68.

Osman, I. (1993). Metastrategy simulated annealing and tabu search heuristic algorithms for the vehicle routing problem. *Annals of Operations Research, 41,* 421-434.

Potvin, J., Kervahut, T., Garcia, B., & Rousseau, J. (1996b). The vehicle routing problem with time windows, part I: Tabu search. *Journal on Computing, 8*(2), 158–164.

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls, *European Journal of Operational Research*, *171*, 750–775.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254-265.

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.

Tavakkoli-Moghaddam, R., Saremi, A. R., & Ziaee, M. S. (2006). A memetic algorithm for a vehicle routing problem with backhauls. *Applied Mathematics and Computation*, *181*, 1049-1060

Thangiah. S. R., Potvin, J., & Sun, T. (1996). Heuristic approaches to vehicle routing with backhauls and time windows, *Computers and Operations Research*, *23*(11), 1043-1057.

Tuntitippawan, N., & Asawarungsaengkul, K. (2016a). An artificial bee colony algorithm for the vehicle routing problem with backhauls and time windows. *International Conference on Industrial Engineering and Operations Management* (p.2788). Kuala Lumpur, Malaysia.

Tuntitippawan, N., & Asawarungsaengkul, K. (2016b). An artificial bee colony algorithm with local search for vehicle routing problem with backhauls and time windows. *KKU Engineering Journal*, *43*, 404-408.

Worawattawechai, T., Intiyot, B., & Jeenanunta, C. (2016a). Heuristic approach to vehicle routing problem with backhauls and time windows. In W. Panichkit kosolkul, & P. Srisuradetchai, (Eds.), *International Conference on Applied Statistics 2016* (pp. 121-128). Phuket, Thailand: Thammasat University.

Worawattawechai, T., Intiyot, B., & Jeenanunta, C. (2016b). Cuckoo search algorithm for the vehicle routing problem with backhauls and time windows. *Panyapiwat Journal*, *8*, 136-149.

Yu, S., Tai, C., Liu, Y., & Gao, L. (2016). An improved artificial bee colony algorithm for vehicle routing problem with time windows: A real case in Dalian, *Advances in Mechanical Engineering, 8*(8), 1–9.