KKU Engineering Journal
https://www.tci-thaijo.org/index.php/kkuenj/index

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

Naritsak Tuntitippawan and Krisada Asawarungsaengkul*

Department of Industrial Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok. 1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok 10800, Thailand.

Received April 2016
Accepted June 2016

Abstract

This paper presents an artificial bee colony algorithm to solve the vehicle routing problem with backhauls and time windows (VRPBTW). This problem is a combination of the vehicle routing problem with backhauls (VRPB) and the vehicle routing problem with time windows (VRPTW). In VRPBTW, a homogenous fleet of vehicles are utilized to deliver goods to customers in linehaul set and then to pick up goods from customers in backhaul set. Vehicle capacity, backhaul and time windows are the major constraints for this problem. The objective of VRPBTW is to minimize the sum of route distance that satisfy all constraints. An artificial bee colony (ABC) algorithm with local search procedures are proposed to solve the modified Solomon's VRPTW benchmark problems. The results of computational experiments reveal that the performance of the proposed ABC algorithm is comparable to the other metaheuristics in terms of the quality of solution.

Keywords: Artificial bee colony algorithm, Vehicle routing problem, Backhauls, Time windows, Local search

1. Introduction

The vehicle routing problem with backhaul and time windows (VRPBTW) is a combination of the vehicle routing with backhual (VRPB) and the vehicle routing problem with time windows (VRPTW). In this VRPBTW, a homogeneous fleet of vehicles are utilized to deliver goods from a central depot to customers (linehauls) and then to pick up goods from customers (backhauls) to the depot. This delivery and pick up procedure allows company to fully utilize the fleet of vehicles and leads to reduce the fuel oil consumption. The major constraints are the vehicle capacity and time windows constraint. Each customer must be serviced within specified time interval or time windows between earliest and latest time deadline. All linehaul customers must be done before backhaul customers, because it is inconvenient to rearrange the delivery goods onboard so that new pickup goods will be accommodated. The objective of this VRPBTW is to minimize the sum of route distance to service all customers with vehicle capacity and time windows constraints.

A literature on the heuristic approach to solve VRPBTW was proposed by [1]. The route construction and heuristic were tested with the modified solomon's benchmark problem. The quality of solution is acceptable. A Genetic algorithm (GA) and a Tabu search heuristic for VRPBTW were performed by [2] and [3], respectively. Both papers used the same benchmark problems as in [1]. A metaheuristic namely a guide local search approach (GLSA) was also done by [4]. GLSA did not perform well comparing to GA in [2].

Recently, a differential evolution algorithm (DEA) for VRPBTW was proposed by [5]. They could find some of the results were better than the best known solution. In additions, [6] proposed a Hybrid metaheuristic algorithm (HMA) which employed simulated annealing and tabu search to solve VRPBTW. The proposed algorithm could also found better solutions than the best known solutions in practical computational time. The performance of HMA in [6] was superior to that of DAE in [5]. Since this VRPBTW is a NP hard problem, we propose an artificial bee colony (ABC) algorithm with the local search procedure to solve it. The Solomon's benchmark problems of which their optimal solutions are known in [7] are used to evaluate the performance of the ABC algorithm in solving the VRPBTW.

2. Problem definition

The VRPBTW is a variant of the vehicle routing problem (VRP). In VRPBTW, there are two subsets of customers including linehauls and backhauls. A central depot supplied a given quantity of goods to each linehaul customer and a given quantity of goods is picked up from each backhaul customer and then returned to the depot. Each customer's demand cannot be split. The backhauls must be visited after the linehauls in each route.

The characteristics of this VRPBTW are as follows: (1) there is a single depot and a fleet of homogeneous fleet of vehicles, (2) each vehicle services only one route, (3) the linehaul vertices must precede the backhaul vertices on each

*Corresponding author. Tel.: +66 2555 2000
Email address: krisadaa@kmutnb.ac.th; a.krisada@gmail.com
doi: 10.14456/kkuenj.2016.141

1. Generate a set of initial food sources (or initial solutions) X_i .
2. Evaluate the fitness $f(X_i)$ for each food source.
3. Set $v = 0$ and $l_i = 0$, $i = 1, 2, ..., Nb$
4. **While** ($v <= MaxIteration$) do

   For each food source X_i , generate X_i' from X_i by using a local search operation

   If $f(X_i') > f(X_i)$ **Then** $X_i = X_i'$ and $l_i = 0$, **Else** $l_i = l_i + 1$ **End If, Next**

   Set $f(\hat{X}) = 0$, F=$\varnothing$

   For each onlooker

   Select food source X_i by using roulette wheel selection method and generate X_i' from X_i by using a local search operation.

   If $f(X_i') > f(\hat{X})$ **Then** $\hat{X} = X_i'$ **End If, Next**

   For each food source X_i , **If** $f(X_i) < f(\hat{X})$ **Then** $i \in F$ **End If, Next**

   Set $\tilde{X} = X_j$ where $j = \arg[\max_{j \in F} \{l_j\}]$

   For each food source X_i , **If** $X_i = \tilde{X}$ **Then** Replace X_i with $\hat{X}$ and $l_i = 0$ **Else** $l_i = l_i + 1$ **End If, Next**

   For each food source X_i , **If** $l_i = limit$ **Then** generate X_i' from X_i by using a local search operation and replace X_i with X_i' **End if**, **Next**

   $v = v + 1$

   Loop

Figure 1 Artificial bee colony algorithm for solving the VRPBTW

route, (4) the time of beginning of service at each vertex must be within to the lower bound and upper bound of time windows of each customer $[a_i, b_i]$. The mathematical modeling of this VRPBTW can be found in [5-6]. The objective of this VRPBTW is to minimize the sum of route distances to service all customers with vehicle capacity and time windows constraints.

3. The proposed algorithms

This section describes the artificial bee colony procedure, generation of initial solution and local search operation. The local search namely λ -interchange which try to exchange customer nodes between 2 routes.

3.1 Artificial bee colony algorithm

The artificial bee colony algorithm is an evolutionary algorithms inspired by behavior of honey bees that try to search for the food or nectar sources around their hive. The search strategy of ABC algorithm starts the initial solutions so called food sources and then groups of bees try to exploit the food sources to obtain the best nectar quantity. The ABC algorithm classifies bees into three types [8] which are employed bee employees, onlookers and scouts. Employed bees exploit the available food sources and gather required information. This information is shared to onlookers and the onlookers select the existing food source by utilizing the roulette wheel selection method to explore the further food source. Employed bees can abandon the old food source when the onlookers can find the best food source. In this case, the employed associated with the old food sources will be assigned to the best food source found by onlookers. However, any food sources will also be abandoned if the quality of food source is not improved for a limit successive iterations, *limit*. After that, the employed bee becomes a scout to look for new food source randomly.

The overall ABC algorithm adapted from [8] can be summarized as in Figure 1. The fitness function $f(X_i)$ is equal to$1/Z(X_i)$; where $Z(X_i)$ is the sum of route distances of the food source X_i. The lower the sum of route distances it is, the more fitness it obtains.

3.2 Initial solutions

The initial solution is generated based on the time-oriented nearest neighbor heuristic proposed by Solomon (1987) [9]. Solomon's nearest neighbor heuristic considers both graphical and temporal closeness of customer. The heuristic starts every route by searching the unrouted customer which is closest to the depot. Next, unrouted customers are sequentially added to current routes or depot. Every unrouted customer who is feasible to be added to routes has to satisfy the time windows, capacity and backhaul constraint.

Let the last customer on the partial route be customer *i* and let *j* denote any unrouted customers who are next visit. The algorithm finds the best unrouted customer to be assigned to the best route by utilizing the cost function, $cost_{ij}$, to select the best unrouted customer [10]. This cost function can be determined from three factors: (1) the direct distance between two customers, d_{ij}; (2) the urgency of delivery from customer *i* to customer *j*, u_{ij} and (3) the waiting time which is the time remaining until the vehicle's last possible service start, w_{ij} . The formulation of three factors and the cost function are as follows:

$$d_{ij} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2} \quad (1)$$

where (x_i, y_i) is geometric location of customer *i;*

$$u_{ij} = b_j - (\mathrm{T}_i + s_i + t_{ij}) \quad (2)$$

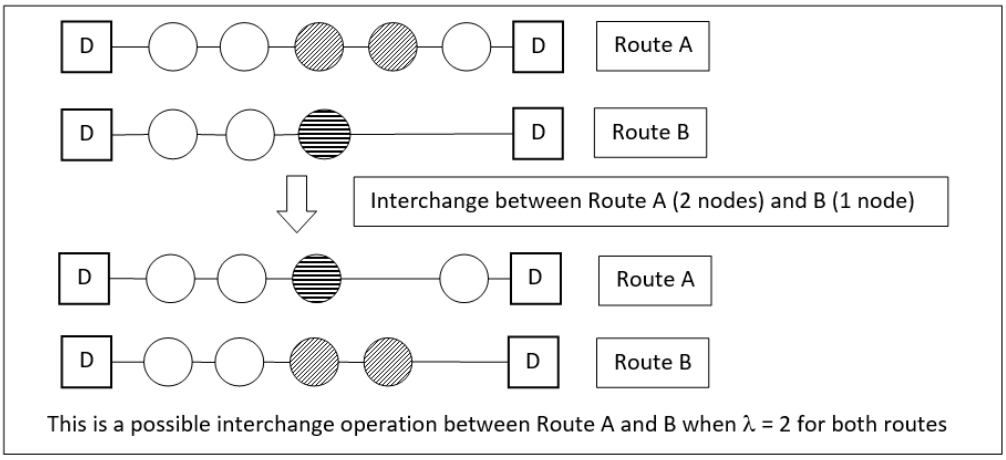

Figure 2 A possible interchange of nodes between two routes when λ = 2 for both routes

Analysis of Variance for Avg. Distance (coded units)for 25 customers

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Number of bees	1	2.040	2.040	2.0402	7.06	0.057
Lambda	1	2.040	2.040	2.0402	7.06	0.057
Number of bees*Lambda	1	2.040	2.040	2.0402	7.06	0.057
Residual Error	4	1.155	1.155	0.2888		
Total	7	7.276				

R-Sq = 84.12% R-Sq(adj) = 72.21%

Analysis of Variance for Avg. Distance (coded units) for 50 customers

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Number of Bees	1	793.21	793.21	793.2	3.09	0.154
Lambda	1	745.37	745.37	745.4	2.90	0.164
Number of Bees*Lambda	1	1601.21	1601.21	1601.2	6.23	0.067
Residual Error	4	1027.32	1027.32	256.8		
Total	7	4167.12				

R-Sq = 75.35% R-Sq(adj) = 56.86%

Analysis of Variance for Avg. Distance (coded units) for 100 customers

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Number of bees	1	0.000	0.000	0.000	0.00	0.996

Figure 3 Results from DOE of ABC-II for problem size of 25, 50, and 100 customers

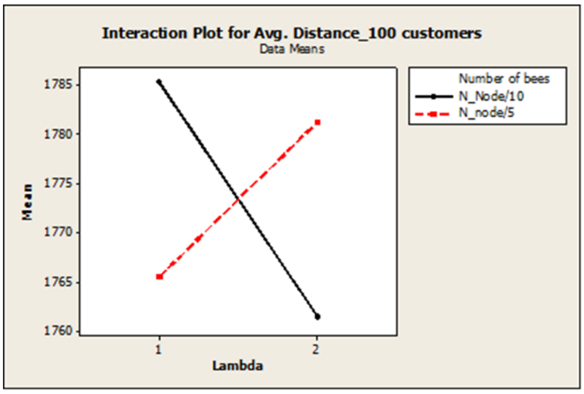

Figure 4 Interaction plot for average distance of problem of 100 customers

where T_i denotes the service start time of customer, s_i is defined as the service time of customer i, t_{ij} denotes travel time between customer i's location to customer j's location;

$$w_{ij} = \max\{0, a_j - (T_i + s_i + t_{ij})\} \tag{3}$$

$$\theta_d d_{ij} + \theta_u u_{ij} + \theta_w w_{ij} \tag{4}$$

where $\theta_d, \theta_u, \theta_w$, are the weight of distance, urgency, and waiting time, respectively.

The lower value of $cost_{ij}$ is more preferable. To generate different initial food sources, the initial weight of distance, urgency, and waiting time are randomly search from range 1 to 15.

3.3 The λ -interchange local search method

The interchange local search method introduced by [11] is a local search that based on λ parameter. Two routes are randomly selected and then the interchange operation is applied. In this paper, λ is equal to 2 for both selected routes that means the maximum number of customers be interchanged is 2 nodes. There are two procedures of local search in this study including:

(1) *One-time* λ *-interchange* referred to **ABC-I** which make the interchange between routes only one time in each operation and

(2) *Best-improvement* λ *-interchange* referred to **ABC-II** which utilizes all possible λ -interchanges for improving the selected routes and then selects the best improvement.

An example of λ -interchange procedure used in this paper is illustrated in Figure 2.

4. Experiment results and discussions

The ABC algorithm with the nearest neighbor heuristic and the λ -interchange were tested with the modified Solomon's benchmark problems. First five problems of Solomon's R1-type data set were modified by randomly selecting 10%, 30%, and 50% of demand nodes to be backhaul customers without changing other characteristics of problem.

The parameters for ABC-I comprise of number of bees, *limit,* and maximum iteration; while those for ABC-II consist of number of bees, *limit*, maximum iteration, and λ (Lambda).

Since there are two local search procedures, we define the parameters for ABC-I based on the past experiment. However, for ABC-II, three design of experiments (DOE) are conducted to determine the appropriate parameters. Number of bees, *limit*, and λ (Lambda) are the factors for the 2^k full factorial design. The levels of each factor are in Table 1. The maximum iteration is fixed at 40,000. The number of customers is denoted by N_node. The response variable of DOE is the sum of route distance.

Table 1 Factor and level of factors of DOE for ABC-II

Parameter	Low (-1)	High (1)
Number of bees	N_node/10	N_node/2
limit	5N_node	10N_node
Lambda	1	2

The results of DOE for problem size of 25 (R105c), 50 (R105c), and 100 (R102c) customers are depicted in Figure 3. It is found that the number of bees and λ are the significant factors affecting to the sum of route distance. The interaction plot of the DOE for problem of 100 customers is displayed in Figure 4. The suitable parameter for ABC-II is in the circle. Thus, the proper parameters of ABC-I and ABC-II can be summarized as in Table 2.

Table 2 Parameter for ABC Algorithm

Parameter	ABC-I	ABC-II
Number of bees	100	N_node/10
Limit	1,000	5N_node#
λ	NA	2
Maximum iteration	20,000	40,000

N_node is number of customer nodes, NA: Not available.

The computational experiments for all problem sets are depicted in Table 3-5. HMA, GA, and DEA in Table 3-5 refer to the algorithms used in [6], [2], and [5], respectively. In these comparisons, the $\%Gap_{opt}$ can be written as

$$\%Gap_{opt} = \frac{metaheuristic\ solution - optimal\ solution}{optimal\ solution} \times 100\%\ .$$

For ABC-I and ABC-II, each problem is solved in five replicates. The best solutions and % coefficient of variation (%CV) are shown in these table. The ABC-II obviously yields better solutions than ABC-I for the problem of 50 and 100 customers. The solutions obtained from ABC-II are comparable to those from HMA and GA; while DEA seems to be the worst. The average computational time of ABC-II for 25 and 50 customers are 11.85, 117.33, and 261.10 (seconds); respectively; while, the average solving time of

Table 3 Computational results for 25 customers

Problem	Optimal solution	ABC-I		ABC-II			HMA*	GA*	DEA*	%Gap$_{Optimal}$			
		Dist.	NV	Dist.	NV	%CV	Dist.	Dist.	Dist.	ABC-II	HMA	GA	DEA
R101a	643.4	643.4	9	643.4	9	0.00	643.4	643.4	643.4	0.00%	0.00%	0.00%	0.00%
R101b	711.1	721.8	11	721.8	10	0.00	721.8	721.8	721.8	1.50%	1.50%	1.50%	1.50%
R101c	674.5	676.8	10	676.8	10	0.00	676.8	682.3	676.8	0.34%	0.34%	1.16%	0.34%
R102a	563.5	563.5	7	563.5	7	0.00	563.5	563.5	565.3	0.00%	0.00%	0.00%	0.32%
R102b	622.3	628.1	9	628.1	9	0.00	628.1	622.3	629.0	0.93%	0.93%	0.00%	1.08%
R102c	584.4	584.4	9	584.4	8	0.00	584.4	584.4	585.3	0.00%	0.00%	0.00%	0.15%
R103a	476.6	476.6	6	476.6	5	0.00	478.8	476.6	489.0	0.00%	0.46%	0.00%	2.60%
R103b	507.0	507.0	7	507.0	7	0.00	507.0	507.0	510.9	0.00%	0.00%	0.00%	0.77%
R103c	475.6	483.0	6	483.0	6	0.00	483.0	483.0	495.0	1.56%	1.56%	1.56%	4.08%
R104a	452.5	453.8	5	453.8	5	0.00	453.8	452.8	459.1	0.29%	0.29%	0.07%	1.46%
R104b	467.6	468.5	6	468.5	6	0.66	468.5	468.5	469.6	0.19%	0.19%	0.19%	0.43%
R104c	446.8	447.7	5	446.8	5	0.00	446.8	446.8	458.7	0.00%	0.00%	0.00%	2.66%
R105a	565.1	565.1	7	565.1	7	0.00	565.1	565.1	565.1	0.00%	0.00%	0.00%	0.00%
R105b	623.5	623.5	8	628.0	8	0.26	623.5	630.2	630.2	0.72%	0.00%	1.07%	1.07%
R105c	591.1	591.1	8	591.1	8	0.00	592.1	592.1	598.5	0.00%	0.17%	0.17%	1.25%

Dist.: the best distance; NV: number of vehicle; %CV: % coefficient of variation; HMA: hybrid metaheuristic algorithm

Table 4 Computational results for 50 customers

Problem	Optimal* solution	ABC-I		ABC-II			HMA	GA	DEA	%Gap$_{Optimal}$			
		Dist.	NV	Dist.	NV	%CV	Dist.	Dist.	Dist.	ABC-II	HMA	GA	DEA
R101a	1122.3	1137.0	15	1134.0	15	0.15	1135.8	1138.1	1138.3	1.04%	1.20%	1.41%	1.43%
R101b	1191.5	1210.4	16	1191.6	16	0.61	1191.6	1192.7	1245.8	0.01%	0.01%	0.10%	4.56%
R101c	1168.6	1183.9	16	1183.9	16	0.70	1183.9	1183.9	1183.9	1.31%	1.31%	1.31%	1.31%
R102a	974.7	1003.8	12	976.5	12	0.21	976.8	976.8	978.7	0.18%	0.22%	0.22%	0.41%
R102b	1024.8	1058.9	14	1054.6	14	0.09	1046.0	1029.2	1046.0	2.91%	2.07%	0.43%	2.07%
R102c	1057.2	1072.3	14	1059.7	14	0.07	1061.6	1059.7	1153.0	0.24%	0.42%	0.24%	9.06%
R103a	811.4	813.0	9	821.6	9	0.45	815.5	813.3	831.1	1.26%	0.51%	0.23%	2.43%
R103b	882.8	902.0	11	887.1	11	1.17	889.3	892.7	895.1	0.49%	0.74%	1.12%	1.39%
R103c	882.1	907.8	11	885.1	10	0.27	887.7	885.5	887.7	0.34%	0.63%	0.39%	0.63%
R104c	733.6	751.8	8	739.3	8	0.23	738.2	741.4	742.2	0.78%	0.63%	1.06%	1.17%
R105a	970.6	993.8	12	985.2	11	0.63	978.5	1002.5	972.8	1.50%	0.81%	3.29%	0.23%
R105b	1007.5	1049.4	13	1024.7	12	0.85	1026.7	1047.8	1030.0	1.71%	1.91%	4.00%	2.23%
R105c	993.4	1018.3	11	993.4	11	0.27	996.2	1018.0	1022.2	0.00%	0.28%	2.48%	2.90%

* This table displays only the problems of which their optimal solutions are known.

Table 5 Computational results for 100 customers

Problem	Optimal* solution	ABC-I		ABC-II			HMA	GA	DEA	%Gap$_{Optimal}$			
		Dist.	NV	Dist.	NV	%CV	Dist.	Dist.	Dist.	ABC-II	HMA	GA	DEA
R101a	1767.9	1830.9	24	1818.6	24	0.42	1811.6	1815.0	1811.6	2.87%	2.47%	2.66%	2.47%
R101b	1877.6	1996	25	1904.5	25	0.46	1891.1	1896.6	1925.9	1.43%	0.72%	1.01%	2.57%
R101c	1895.1	2006.6	26	1928.2	24	0.40	1911.2	1905.9	1930.2	1.75%	0.85%	0.57%	1.85%
R102a	1600.5	1692.7	21	1640.7	21	0.48	1623.7	1622.9	1649.8	2.51%	1.45%	1.40%	3.08%
R102b	1639.2	1719.6	22	1717.3	22	0.48	1724	1688.1	1758.2	4.76%	5.17%	2.98%	7.26%
R102c	1721.3	1807.4	23	1752.2	21	0.46	1759.8	1735.7	1777.1	1.80%	2.24%	0.84%	3.24%

HMA for 25, 50, and 100 customers are 1.57, 15.91, and 100.17 (seconds), respectively. However, the computational time of ABC-II is practically acceptable. Moreover, the overall %CV of ABC-II are lower than 1% which indicate that the ABC-II is capable to yield the solutions with low variation.

5. Conclusions

The proposed ABC algorithm with the λ-interchange local search are examined with the modified Solomon's benchmark problems. It can be concluded that the performance of the proposed ABC algorithm is comparable to HMA and GA. Moreover, it is superior to DEA. However, the improvement on the ABC algorithm to solve the large-scale problem may be done for the future work by improving the route construction, since the number of vehicles required for some problems are still higher than that of other methods.

6. Acknowledgements

This research was funded by Faculty of Engineering, King Mongkut's University of Technology North Bangkok, Thailand. This support is gratefully acknowledged.

7. References

[1] Thangiah SR, Potvin JY, Sun T. Heuristic approaches to vehicle routing with backhauls and time windows. Computers & Operations Research 1996;23:1043-1057.

[2] Potvin JY, Duhamel C, Guertin F. A genetic algorithm for vehicle routing with backhauling. Applied Intelligence 1996;6:345-355.

[3] Duhamel C, Potvin JY, Rousseau JM. A tabu search heuristic for the vehicle routing problem with backhauls and time windows. Transportation Science 1997;31(1):49-59

[4] Zhong Y, Cole MH. A vehicle routing problem with backhauls and time windows: a guided local search solution. Transportation Research Part E 2005;41:131-144.

[5] Kucukoglu I, Ozturk N. A differential evolution approach for the vehicle routing problem with backhauls and time windows. Journal of Advanced Transportation 2014;48:942-956.

[6] Kucukoglu I, Ozturk N. An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows. Computers & Industrial Engineering 2015;86:60-68.

[7] Gélinas S, Desrochers M, Desrosiers J, Solomon MM. A new branching strategy for time constrained routing problems with application to backhauling. Annals of Operations Research 1995;61(1):91-109.

[8] Szeto WY, Wu Y, Ho SC. An artificial bee colony algorithm for the capacitated vehicle routing problem. European Journal of Operational Research 2011;215(1):126-135.

[9] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time windows constraints. Operations Research 1987;35:254-265.

[10] Pang KW. An adaptive parallel route construction heuristic for the vehicle routing problem with time windows constraints. Expert Systems with Applications 2011; 38:11939-11946.

[11] Osman IH, Christofides N. Capacitated clustering problems by hybrid simulated annealing and tabu search. International Transactions in Operational Research 1994;1(3):317-336.