Bees Algorithm for Vehicle Routing Problems with Time Windows

Malek Alzaqebah, Sana Jawarneh, Hafiz Mohd Sarim, and Salwani Abdullah

***Abstract*—This paper presents the bees algorithm for vehicle routing problems within time windows (VRPTW). The VRPTW aims to determine the optimal route for a number of vehicles when serving a set of customers within a predefined time interval (the time window). The objective in VRPTW is to minimize overall transportation cost. Various heuristic and metaheuristic approaches have been developed in literature to produce high-quality solutions for this problem because of its high complication rate and extensive implementation in real-life applications. This research investigates the use of bee algorithms (BA) for VRPTW and identifying the strengths and weaknesses.**

***Index Terms*—Foraging behaviour, bees algorithm, vehicle routing problem with time windows.**

I. Introduction

Vehicle routing problems (VRP) have an important role in the domains of transportation, delivery, and logistics. Hence, numerous research works have been undertaken to study VRP since 1959 [1]. The work presented by Dantzig and Ramser in 1959 proposed the problem as a generalized Travelling Salesman Problem (TSP). Later, a huge number of studies have been conducted on number of VRP divisions (e.g. VRP with Time Windows (VRPTW), VRP with Pick-Up and Delivery (VRPPD), Multiple Depot VRP (MDVRP), MDVRP with Time Windows (MD-VRPTW), and Capacitated VRP (CVRP)).

VRPTW is an NP-hard problem, which is concerned with determining the best routing of a set of limited capacity vehicles between a central depot and a number of scattered customers, where customers must be visited within predefined time duration (the time window).

Several approaches have been developed for VRPTW. Solomon [2] first introduced heuristics to solve this problem. In recent years, metaheuristics has become increasingly popular. Metaheuristics can be classified either as single-based approaches such as tabu search and simulated annealing, or population-based approaches such as ant colony optimization and genetic algorithms [3].

Population-based approaches are classified into two categories, which are evolutionary algorithms and swarm intelligence algorithms [4]. This classification is based upon natural phenomena being modeled by the algorithms. Evolutionary algorithms apply the theory of evolution to create new species [3]. By contrast, swarm intelligence (SI) algorithms rely on metaheuristics that mimic collective behavior of problem-solving processes in self-organized systems [5]. The interactions between agents in social colonies with their environment leads to the collective intelligence [6]. SI features have inspired some researchers to develop algorithms for VRPTW, including the ant colony optimization algorithms [7], bee colony optimization (BCO) algorithms [8] and the artificial bee colony (ABC) algorithms [9], [10], to name a few examples.

With regards to swarm intelligence, researchers are concerned with developing algorithms that model the behavior of honeybees. Honeybee algorithms are categorised based on the behavior of bees [11]: queen, marriage, and foraging bee. Examples of foraging behavior algorithms can be found in ABC [12], bees algorithm (BA) [13], and bee colony optimization (BCO) [14], [15]. In this paper, Bees algorithm (BA) is investigated for VRPTW and some strengths and weaknesses are identified.

The remainder of the paper is organized as follows: in the next section VRPTW is described and its formulation with some relevant-literature review is provided. In the third section, the proposed algorithm is presented. The fourth section provides an extensive computational study comparing the results of BA and with the best-known approaches. Finally, the paper is concluded and some future research directions are listed.

II. The Vehicle Routing Problems with Time Windows

This study focuses on the VRPTW in order to understand its complexity and maximize its advantages in real life. In recent years, the VRPTW has gained considerable research attention in network optimization [16].

Ellabib *et al.* [17] defined the VRPTW as a generalisation of the VRP. The VRPTW often appears in many real applications. VRPTW can be also used to achieve the following objectives: optimize the number of vehicles that must serve a set of customers, to determine the customers that each vehicle should serve, and to determine the order of cost that should be minimised as much as possible. The VRPTW is subject to capacity of the vehicle and service time constraints. The problem contains vehicle tasks that minimise task cost.

Manuscript received March 15, 2018; revised May 2, 2018.

M. Alzaqebah is with the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia (e-mail: Malek.zaqeba@gmail.com).

S. Jawarneh is the Department of Computer Science, Community College Dammam, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia (e-mail: sana_jawarneh@yahoo.com).

H. M. Sarim and S. Abdullah are with Data Mining and Optimisation Research Group, Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia (e-mail: hms@ukm.edu.my, salwani@ukm.edu.my).

doi: 10.18178/ijmlc.2018.8.3.693

According to the definition above, the VRPTW processes a specific number of requests to be visited by a certain number of vehicles. All customers are supposed to require a limited quantity of demands. Furthermore, each route must start and finish at depot, and each customer must be visited only once.

The main formulation of the VRPTW was proposed in 1987 by Solomon [2]. The standard benchmark of Solomon's datasets contains 56 instances, and every instance has 100 customers. In the network, each vehicle represents one route. The vehicle starts routing from the departure point and arrives at specific destinations (customers), then returns to the departure point. In considering Solomon's benchmark, each route has its own cost c_{ij} and the traveling time t_{ij}. Each vehicle spends one-time slot (time unit) to travel one distance unit, so the speed of the vehicle is assumed to be constant.

The time window boundaries are defined by the earliest and latest arrival times (the time interval); in which the vehicle must arrive at the customer's place before the latest arrival time. The vehicle should wait in cases where it arrives before the earliest arrival time. The service time of the customer must be taking into account, which represents the time that is spent to load or unload demands. The demands' size is considered unique for all customers. Route time (time window) is the time taken by vehicles to visit all customers in the same route.

Equation 1 represents the main function for VRPTW [2]:

$$\textit{Minimise} \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{N} c_{ij} \quad x_{ijk} \tag{1}$$

where,

- $x_{ijk} = \begin{cases} 0 & \text{if there is no arc from node } i \text{ to node } j \\ 1 & \text{otherwise} \end{cases} \quad i \neq j, i, j \in \{0,1,\ldots N\}$
- N number of customers
- K number of vehicles
- c_{ij} cost on arc between two customers i *to* j.

III. Bees Algorithm (BA)

BA was firstly proposed by Pham *et al.* [13] and algorithm has been applied in various optimization fields [18]-[22].

BA consists of two groups of bees i.e., scout bees and foragers. The scout bees are involved in searching for food sources (solutions) and recruiting foragers to continue searching in the discovered food sources. The scout bees search for food sources randomly in the search space (Exploration). During the search, they collect useful information and return back to the hive to share the information with other bees in the hive. and the scouts then recruit other bees as foragers to continue searching in previously discovered food sources (Exploitation).

Based on the collected information by the scout bees, the good food sources will have a larger number of recruited bees to search in the neighborhood of these food sources. Using these methods, the bees algorithm offers both global and local search, or exploration and exploitation processes; the search will be performed in existing food sources, including the best sources, and the scout bees performs global random search [13]. The general pseudo-code for the BA is shown in Fig. 1.

```
Initialization:
NumOfIte : maximum number of iteration;
populationsize : #of solutions in the population;
ne:  #of elite solutions;
nre: list of recruited bees assigned to elite solutions;
nb: #of best solutions;
nrb: list of recruited bees assigned to best solutions;
stlim: stagnation iteration limit for abandonment solutions;

Initialise the population; Compute the initial value of the fitness, f(Sol);
iteration ←  0;
best solution, Solbest ← Sol;
Improvement:
do while (iteration < NumOfIte)
   Rank all solutions in the population based on fitness value.
     nbSet = the top nb solutions of population;
     neSet = the top ne solutions of nbSet;
       for   (i=1  :  nrb)
         for (j=1  :   nb)
           Sol*  ← neSet_j;
           A random neighbourhood method is applied on Sol* ;
           Update the population;
         end for
       end for
     for (j=1   :    nre)
        for (h=1 :  nrb)
           Sol**  ← neSet_j;
           random neighbourhood method is applied on Sol** ;
        end for
       end for
       Solbest ← the best solution found so far;
       random neighbourhood method is applied on the rest of (nb–ne)
       solutions;
       Update the population;
       iteration++;
end do
```

Fig. 1. Pseudo-code for the BA.

As can be observed by Fig. 1, the algorithm begins with randomly initializing the population, where the number of scout bees is equal to the number of solutions in the population. The scout bees randomly search for the food sources (solutions). Once they find the food sources, they return to the hive and start recruiting other bees to exploit those food sources. Next, all solutions in the population are evaluated and ranked based on a fitness function. A number of *nb* highest ranked solutions are selected for a local search. Then, the scout bees recruit the foragers to search on the neighbourhood of the selected solutions, as follows:

- Every scout bee returned from each *nb* (the best solutions) will recruit the *nest* (the top *nb* solutions from the population) foragers for a local exploration.
- The scout bees that visited the first *ne* elite solutions among the best *nb* solutions will recruit *nre* foragers and apply a random neighbourhood structure.
- The scout bees that visited the rest (*nb*–*ne*) solutions will recruit *nrb* < *nre* foragers and apply a random neighbourhood structure.

The main idea here is to recruit more foragers and apply a random neighborhood structure on elite solutions, which are considered the most promising solutions in the search space.

A. Neighborhood Structures

We considered that the sequence of the customers to be visited (*a*, *b*, *c* and *d*) is {*a*→*b*→*c*→*d*} that exist in same route.

The neighborhood methods is presented as follows:

N1: perform one shift at similar route, example b is moved to the last; the new order will be $\{a \rightarrow c \rightarrow d \rightarrow b\}$.

N2: perform two shifts at similar route, example both a and b are moved to a position after c and d; the new order will be $\{c \rightarrow d \rightarrow a \rightarrow b\}$.

N3: perform one swap at similar route, where a and c are exchanged; the new order will be $\{c \rightarrow b \rightarrow a \rightarrow d\}$.

N4: perform two swaps at similar route, where a and b are exchanged with c and d, the new order will be $\{c \rightarrow d \rightarrow a \rightarrow b\}$.

B. Standard Dataset: Solomon's Benchmark

The standard benchmark datasets for the VRPTW were introduced by Solomon [2], and they can be freely downloaded at: http://neo.lcc.uma.es/vrp/vrp-instances /capacitated-vrp-with-time-windows-instances.

Six benchmark problems exist. Note that in this study, the performance of BA on Solomon's 56 VRPTW 100-customer instances (which have been widely used by different studies throughout the years) is discussed. This benchmark is explained as follows:

The problems differ in the total number of vehicles, travelling time of vehicles, vehicle capacity and the place of the customers, and their suitable time to serve. In addition, each customer has a preferred time window and details, such as location given in the form of x and y coordinates, quantity of demand, ready time, service time, and due time needed.

All the instances have 100 customers, which are generally assumed as the problem size for rendering comparisons in VRPTW. The time expended to travel between the customers is equal to the Euclidean distance. The 56 instances are separated to six categories based on the preparation of customers' places and the time windows. These six categories are named as R1, R2, C1, C2, RC1, and RC2. The category with letter R means that customers are placed remotely in the instances Therefore, the travelling distances between customers are somehow long. While the category with letter C contains the instances with clustered customers, in this category the places of the customers are close to each other, so the travelling distances somehow short. The category with letters R and C represents the instances with combination of remotely placed and clustered customers [2].

IV. Experimental Results

Our experiment results are based on 31 separated runs. The Java programming language was used to code our algorithm based on Intel® Core™ i3 processors. The implementation times were between 30 and 650 seconds, which may differ based on the size of the instance.

TABLE I: Parameters Setting

Parameter	Value
The population size	50
Number of iterations	500
ne= number of elite solutions	2
nre= recruited bees for elite solutions	30
nb= number of best solutions	4
nrb= recruited bees for rest best solutions	10
stlim= stagnation iterations limit for abandonment solutions	10

Table I shows the final parameter values, which have been set experimentally from the average results of 10 runs.

Table II shows the results comparison between our BA and the best-known results in terms of number of vehicles and the total distance.

TABLE II: Comparison between the Best-Known Results and BA

No	Instance	Best-known Result		Source	BA	
		N.V	*Distance*		*N.V*	*Distance*
0	R1-01	18	1607.7	[23]	20	1642.67
1	R1-02	17	1434	[23]	18	1480.73
2	R1-03	13	1175.67	[24]	16	1240.87
3	R1-04	10	982.01	[25]	11	1047.06
4	R1-05	15	1346.12	[26]	16	1369.52
5	R1-06	13	1234.6	[27]	14	1271.13
6	R1-07	11	1051.84	[26]	12	1129.99
7	R1-08	9	960.88	[28]	11	1004.11
8	R1-09	12	1013.2	[29]	14	1170.5
9	R1-10	12	1068	[27]	13	1123.36
10	R1-11	12	1048.7	[27]	13	1101.59
11	R1-12	10	953.63	[25]	11	1019.84
12	R2-01	4	1252.37	[30]	8	1185.57
13	R2-02	3	1158.98	[24]	8	1103.15
14	R2-03	3	939.5	[31]	7	958.94
15	R2-04	2	825.52	[32]	6	818.44
16	R2-05	3	994.42	[33]	7	1020.53
17	R2-06	3	833	[34]	6	960.29
18	R2-07	3	814.78	[25]	4	905.7
19	R2-08	2	726.75	[35]	5	764.9
20	R2-09	3	855	[34]	6	943.16
21	R2-10	3	939.34	[35]	7	1003.91
22	R2-11	2	885.71	[31]	5	837.66
23	C1-01	10	827.3	[23]	10	828.94
24	C1-02	10	827.3	[23]	11	828.94
25	C1-03	10	826.3	[36]	10	828.94
26	C1-04	10	822.9	[36]	10	858.9
27	C1-05	10	827.3	[36]	11	828.94
28	C1-06	10	827.3	[23]	10	828.94
29	C1-07	10	827.3	[36]	11	828.94
30	C1-08	10	827.3	[36]	10	828.94
31	C1-09	10	827.3	[36]	10	828.94
32	C2-01	3	589.1	[27]	3	591.56
33	C2-02	3	589.1	[27]	3	591.56
34	C2-03	3	591.17	[47]	3	600.54
35	C2-04	3	590.6	[36]	4	610.01
36	C2-05	3	588.88	[36]	3	588.88
37	C2-06	3	588.49	[24]	3	588.88
38	C2-07	3	588.29	[25]	3	589.58
39	C2-08	3	588.32	[25]	3	591.65
40	RC1-01	15	1619.8	[37]	16	1634.52
41	RC1-02	13	1530.86	[38]	15	1492.89
42	RC1-03	11	1261.67	[39]	12	1334.57
43	RC1-04	10	1135.48	[40]	11	1215.62
44	RC1-05	13	1629.44	[41]	17	1546.43
45	RC1-06	12	1395.4	[29]	13	1423.1
46	RC1-07	11	1230.5	[42]	12	1300
47	RC1-08	10	1139.8	[42]	12	1193.68
48	RC2-01	4	1249	[34]	9	1308.76
49	RC2-02	4	1164.3	[42]	8	1167
50	RC2-03	3	1049.62	[43]	6	1014.79
51	RC2-04	3	798.41	[35]	4	881.88
52	RC2-05	4	1297.19	[35]	8	1210.68
53	RC2-06	3	1146.32	[45]	6	1112.38
54	RC2-07	3	1061.14	[45]	6	1059.62
55	RC2-08	3	828.14	[46]	7	882.06

Note: N.V = number of vehicles.

From the results in Table II, we can observe that the algorithm is able to obtain good results. This may due to the method that has the scout bees to explore the search space and they recruit other bees for exploitation and local search. Nevertheless, some of the datasets are not able to obtain comparable results, which may be due to the number of parameters used in the algorithm that needs to be carefully tuned.

V. CONCLUSION

The proposed Bees algorithm for the VRPTW is able to perform good quality solutions. Experimental results show that the Bees algorithm can achieve comparable results when we compare it with state-of-the-art approaches from the literature, bearing in mind the overall traveling distance is our main objective to be minimized for VRPTW.

The main strength of BA is that the algorithm has both global exploration performed by scout bees and local exploitation performed by recruiter bees. On the other hand, the main weakness of the algorithm is that it is parameter dependent, so each instance may require different parameters values. Using automatic parameter tuning strategies is part of the planned implementation in our future work.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education, Malaysia (FRGS/1/2015/ICT02/UKM/01/2) and the Universiti Kebangsaan Malaysia (DIP-2016-024).

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, no. 1, pp. 80-91, 1959.

[2] M. M. Solomon, “Algorithms for the vehicle routing and scheduling problems with time window constraints,” *Operations Research*, vol. 35, no. 2, 254-265. 1987.

[3] E. G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74, John Wiley & Sons. 2009.

[4] J. Dreo, A. Petrowski, P. Siarry, and E. Taillard, “Metaheuristics for hard optimization: methods and case studies,” *Springer Science & Business Media*, Berlin (Heidelberg): Springer-Verlag, 2006.

[5] M. Farooq, “Bee-inspired protocol engineering: From nature to networks,” Springer, Berlin, Heidelberg, Germany, 2008.

[6] A. C. Kamil, J. R. Krebs, and R. Pulliam, *Foraging Behavior*, Plenum Press, New York and London. 1987.

[7] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E. G. Talbi, “A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests,” *Applied Soft Computing*, vol. 12, no. 4, pp. 1426-1439, 2012.

[8] S. Jawarneh and S. Abdullah, “Sequential insertion heuristic with adaptive bee colony optimisation algorithm for vehicle routing problem with time windows,” *PloS One*, vol. 10, no. 7, p.e0130224. 2015.

[9] Y. J. Shi, F. W. Meng, and G. J. Shen, “A modified artificial bee colony algorithm for vehicle routing problems with time windows,” *Information Technology Journal*, vol. 11, no. 10, p. 1490, 2012.

[10] M. Alzaqebah, S. Abdullah, and S. Jawarneh, “Modified artificial bee colony for the vehicle routing problems with time windows,” *Springer Plus*, vol. 5, no. 1, p. 1298, 2016.

[11] A. Baykasoglu, L. Ozbakir, and P. Tapkan, “Artificial bee colony algorithm and its application to generalized assignment problem. Swarm intelligence: Focus on ant and particle swarm optimization,” p.p. 113-144, 2007.

[12] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, vol. 200, 2005.

[13] D.T. Pham, A. Ghanbarzadeh, E. Koc, S.Otri, S. Rahim, and M. Zaidi, “The bees algorithm,” *Technical Note*, Manufacturing Engineering Centre, Cardiff University, UK, pp. 1-57, 2005.

[14] P. Lucic and D. Teodorovic. “Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence,” in *Proc. the TRISTAN IV Triennial Symposium on Transportation Analysis*, pp. 441-445, June 2001.

[15] P. Lucic and D. Teodorovic, “Transportation modeling: An artificial life approach,” in *Proc. 14th IEEE International Conference on Tools with Artificial Intelligence*, 2002, pp. 216-223.

[16] H. K. Rajagopalan, F.E. Vergara, C. Saydam, and J. Xiao, “Developing effective meta-heuristics for a probabilistic location model via experimental design,” *European Journal of Operational Research*, vol. 177, no. 1, pp. 83-101. 2007.

[17] I. Ellabib, O. A. Basir, and P. Calamai, “An experimental study of a simple ant colony system for the vehicle routing problem with time windows,” *Ant Algorithms*, pp. 53-64, Springer Berlin Heidelberg, 2002.

[18] D. T. Pham, E. Koc, A. Ghanbarzadeh, and S. Otri, “Optimisation of the weights of multi-layered perceptrons using the Bees Algorithm,” in *Proc. 5th International Symposium on Intelligent Manufacturing Systems*, Turkey, pp. 38-46, 2006.

[19] D. T. Pham, E. Kog, A. Ghanbarzadeh, S. Otri, S. Rahim, and M. Zaidi, “The Bees Algorithm – A novel tool for complex optimisation problems,” in *Proc. 2nd International Virtual Conference on Intelligent Production Machines and Systems*, Oxford, Elsevier, 2006a, pp. 454-459.

[20] D. T. Pham, S. Otri, A. Ghanbarzadeh, and E. Kog, “Application of the Bees Algorithm to the training of learning vector quantisation networks for control chart pattern recognition,” *ICTTA'06 Information and Communication Technologies*, pp. 1624-1629, 2006b.

[21] D.T. Pham, A. A. Afify, and E. Koc, “Manufacturing cell formation using the Bees Algorithm,” in *Proc. the Third International Virtual Conference on Intelligent Production Machines and Systems*, Whittles, Dunbeath, Scotland, 2007, pp. 523-528.

[22] D. T. Pham and M. Castellani, “The Bees Algorithm – modelling foraging behaviour to solve continuous optimisation problems,” Proc. ImechE, Part C, vol. 223, pp. 12, pp. 2919-2938, 2009.

[23] M. Desrochers, J. Desrosiers, and M.Solomon, “A new optimization algorithm for the vehicle routing problem with time windows,” *Operations Research*, vol. 40, no. 2, pp. 342-354, April 1992.

[24] H. C. Lau, Y. F Lim, and Q. Liu, “Diversification of neighborhood via constraint-based local search and its application to VRPTW,” in *Proc. CP-AI-OR 2001 Workshop*, 2001.

[25] Y. Rochat and É. D. Taillard, “Probabilistic diversification and intensification in local search for vehicle routing,” *Journal of Heuristics*, Sep. 1, vol. 1, no. 1, pp. 147-167, 1995.

[26] B. Kallehauge and K. J. Larsen. “Lagrangean duality applied on vehicle routing with time windows,” *Informatics and Mathematical Modelling (IMM)*, Technical University of Denmark, 2001.

[27] W. Cook, and J. L. Rich, “A parallel cutting-plane algorithm for the vehicle routing problem with time windows,” Technical Report, Computational and Applied Mathematics Department, Rice University, Houston, TX, May 28, 1999.

[28] O. Bräysy and M. Gendreau, “Metaheuristics for the vehicle routing problem with time windows,” Report STF42 A, 2001.

[29] W. C. Chiang and R. A. Russell, “A reactive tabu search metaheuristic for the vehicle routing problem with time windows,” *INFORMS Journal on Computing*, vol. 9, no. 4, pp. 417-430, 1997.

[30] J. Homberger and H. Gehring, “Two evolutionary metaheuristics for the vehicle routing problem with time windows,” Infor., , vol. 37, no. 3, p. 297 Aug. 1, 1999.

[31] M. Woch and P.Łebkowski. “Sequential simulated annealing for the vehicle routing problem with time windows,” *Decision Making in Manufacturing and Services*, 2009, vol. 3, no. 1-2, pp. 87-100, 2009.

[32] R. Bent and P. Van Hentenryck, “A two-stage hybrid local search for the vehicle routing problem with time windows,” T*ransportation Science*, vol. 38, no. 4, pp. 515-530, Nov. 2004.

[33] L. M. Rousseau, M. Gendreau, and G. Pesant, “Using constraint-based operators to solve the vehicle routing problem with time windows,” *Journal of Heuristics*, vol. 8, no. 1, pp. 43-58, 2002.

[34] S. R. Thangiah, I. H. Osman, and T. Sun, “Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows,” Technical Report SRU CpSc-TR-94-27, Computer Science Department, Slippery Rock University, 1994.

[35] D. Mester, O. Bräysy, and W. Dullaert, “A multi-parametric evolution strategies algorithm for vehicle routing problems,” E*xpert Systems with Applications*, vol. 32, no. 2, pp. 508-517, 2007.

[36] J. Tavares, P. Machado, F. B. Pereira, and E. Costa, “On the influence of GVR in vehicle routing,” in *Proc. the 2003 ACM Symposium on Applied Computing*, pp. 753-758, ACM, 2003.

[37] N. Kohl, J. Desrosiers, O. B. Madsen, M. M. Solomon, and F. Soumis, “2-path cuts for the vehicle routing problem with time windows,” *Transportation Science*, vol. 33, no. 1, 101-16, 1999.

[38] R. Cordone and R. W. Calvo. “A heuristic for the vehicle routing problem with time windows,” *Journal of Heuristics*, vol. 7, no. 2, 2001, pp. 107-129.

[39] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," *Principles and Practice of Constraint Programming*, Springer Berlin Heidelberg, pp. 417-431.
[40] J. F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928-936, 2001.
[41] D. B. Backer, V. Furnon, P. Shaw, P. Kilby, and P. Prosser, "Solving vehicle routing problems using constraint programming and metaheuristics," *Journal of Heuristics*, vol. 6, no. 4, pp. 501-523, 2000.
[42] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," *Transportation Science*, vol. 31, no. 2, pp. 170-186, May 1997.
[43] Z. J. Czech and P. Czarnas, "Parallel simulated annealing for the vehicle routing problem with time windows," *Ineuromicro-pdp*, p. 0376, 2002.
[44] D. I. Eine Homberger, "Verteilt-parallele Metaheuristik," *Verteilt-parallele Metaheuristiken zur Tourenplanung*, pp. 139-165, Deutscher Universitätsverlag, 2000.
[45] Z. J. Czech and P. Czarnas. "Parallel simulated annealing for the vehicle routing problem with time windows," Ineuromicro-pdp, 2002.
[46] T. Ibaraki, M. Kubo, T. Masuda, T. Uno, and M. Yagiura, "Effective local search algorithms for the vehicle routing problem with general time window constraints," in *Proc. MIC*, 2001, pp. 293-297.

Malek Alzaqebah obtained his BSc in computer science from Al-Balqa Applied University, Jordan and his master degree specialising in information technology from Universiti Utara Malaysia (UUM). He did his PhD in computer science at Universiti Kebangsaan Malaysia (UKM). Now, he is an assistant professor at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interest involves metaheuristic algorithms for combinatorial optimisation problems such as university timetabling, job shop scheduling, vehicle routing and search based software engineering.

Sana Jawarneh obtained her BSc in computer engineering from Yarmouk University and her PhD in computer science at Universiti Kebangsaan Malaysia. Now, she is an assistant professor at Department of computer science, Community college, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interest falls under meta-heuristic algorithms in various optimisation problems.

Hafiz Mohd Sarim is a senior lecturer at the Center for Artificial Intelligence Technology, located at Universiti Kebangsaan Malaysia. His primary research areas are in machine learning and artificial intelligence in games, with a particular interest in emotional agents and affective decision making.

Salwani Abdullah obtained her BSc in computer science from Universiti Teknologi Malaysia and her master degree specialising in computer science from Universiti Kebangsaan Malaysia (UKM). She did her PhD in computer science at University of Nottingham, United Kingdom. Now, she is a professor in computational optimisation at the Faculty of Information Science and Technology, UKM. Her research interest falls under Artificial Intelligence and Operation Research, particularly in meta-heuristic algorithms in optimisation areas that involve different real world applications in static and dynamic optimisation problems.