International Journal of Advances in Computer Networks and Its Security
Volume 6 : Issue 3 [ISSN 2250-3757]
Publication Date : 30 December, 2016

Bee Algorithm for the vehicle routing problems with time windows

Malek Alzaqebah , Salwani Abdullah and Sana Jawarneh

***Abstract*—The natural honeybee's behavior has been modelled by researchers to solve optimization problems. This paper introduces the Bee algorithm based on honeybee behavior for the vehicle routing problem with time windows (VRPTW). The algorithm has been tested on Solomon instances. The encoding of the problem and the operations needed to implement algorithm are outlined. Experiments show that on the considered instances, using BA performs almost as well as planning for long-term, while using much less computation time. Overall, computational investigations show that the proposed algorithm is good and promising approach for the VRPTW.**

Keywords—Foraging behaviour; bee algorithm; vehicle routing problem with time windows.

I. Introduction

This Vehicle routing problem with time windows (VRPTW) is an optimisation problem which has attracted a lot of attention in the literature. The VRPTW is a NP-hard problem which consist of defining a set of routes that satisfy the customers' requests. Each route is performed by a vehicle with a permanent and limited capacity and the total demand (load) should not exceed the vehicle's capacity. The vehicle is only allowed to arrive at a customer's location within a certain time window. The solution to the VRPTW consists of the set of routes with the minimum travelled distance (Desrochers et al., 1988). Number of surveys of existing VRPTW methods and applications can be founded in (Toth and Vigo, 2014), (Kumar et al., 2012), (Bräysy and Gendreau, 2005a, 2005b), (Ioannou et al., 2001), (Toth and Vigo, 2002), and (Solomon and Desrosiers, 1988).

Honeybees algorithms have attracted much attention of researchers over years, which they are classified into three different types based on behaviour (Baykasoglu et al. 2007): foraging, marriage and queen bee. This behaviour simulates the natural behaviour of real honey bee, in the way of finding food and sharing important information about food sources. Examples of Honeybee algorithms that have been applied to the VRPTW are Artificial Bee Colony (Shi et al., 2012, Bhagade and Puranik, 2012) bee colony optimisation (Jawarneh and Abdullah, 2015).

Malek Alzaqebah
Department of Computer Science, Jadara University
Jordan

Salwani Abdullah and Sana Jawarneh
Data Mining and Optimisation Research Group, Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia
Malaysia

Experimental results show that BA is able to obtain good results where the comparison between BA and state-of-the-art approaches indicates that BA able to obtain good enough results, as represented by reducing the distance travelled, which is the main objective of the VRPTW.

This paper is organised as follows: Section II presents the VRPTW and its formulation while Section III introduces the BA. The experimental results are presented and analysed in Section IV and the conclusion is given in Section V.

II. Vehicle Routing Problems with Time Windows (VRPTW)

For VRPTW we considered the Solomon's benchmark, which has 56 instances with 100 customers need to be served by a number of vehicles. Each vehicle starts from the depot, arrives at the customer's location, and returns back to the depot, this direction will be considered as route. Each customer must be visited once by one vehicle, and each vehicle in the problem have the same capacity while the total capacity of all the demands must not exceed the maximum capacity of the vehicle (no overloading). The time window constraints are specified by a pre-defined time interval (the earliest arrival time and latest arrival time), where the vehicles must arrive at the customer's location not later than the latest time otherwise the vehicle must wait. The service time (unique time for unloading/loading regardless of the demands' size).

Solomon (1987) proposes the formulation of the VRPTW. The main function in the problem is illustrated in equation 1:

$$\text{Minimise} \sum_{i=0}^{N} \sum_{j=0, j\neq i}^{N} \sum_{k=1}^{K} c_{ij} \; x_{ijk} \qquad (1)$$

where

$$X_{ijk} = \begin{cases} 0 & \text{if there is no arc from node i to node j} \\ 1 & \text{otherwise} \end{cases} \quad i \neq j,\ i,\ j \in \{0,1,2,\ldots\ldots N\}$$

- N number of customers (0 for the central depot)
- K number of vehicles
- c_{ij} cost acquired from customer i to customer j.

The time windows constraint that should not be violated is presented as follow:

$$\sum_{k=1}^{K}\sum_{i=0,j\neq i}^{N} x_{ijk}\,(t_i + t_{ij} + f_i + w_i) \le t_j \quad \text{for } j \in \{1,2\ldots,N\} \quad (2)$$

$$e_i \le (t_i + w_i) \le l_j \qquad \text{for } i \in \{1,2,\ldots\ldots\ldots,N\} \qquad (3)$$

III. The BEE ALGORITHM (BA)

BA firstly proposed by Pham et al. (2005). BA has been applied in various optimisation fields (Pham et al 2006, 2006a, 2006b, 2007; Pham & Castellani, 2009). 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 recruit foragers are involved in searching on the founded food sources.

The scout bees search for food sources randomly in the search space. Then they return to the hive with the collected information and start recruiting more bees to exploit those food sources. Logically, more recruited bees will be sent to search around the better food sources, where they perform a search similar to a local search. To ensure that the search continues until good food sources are found, including the best, scout bees continue on a global random search, while some bees are recruited to perform a local search (Pham et al 2005).

As shown in Figure 1, the initial population is randomly generated, where the number of solutions in the population is equal to the number of scout bees. 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. The scout bees evaluate the solution based on the fitness function, and then they rank the solutions in the population according to the fitness values.

A number of *nb* with highest ranked solutions are selected for a local exploration. Then, the scout bees recruit the foragers to search on the neighborhood of the selected solutions, deterministically as follows:

- Each scout bee that returned from one of the *nb* 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 neighborhood structure.
- The scout bees that visited the remaining (*nb*–*ne*) solutions will recruit *nrb* < *nre* foragers and apply a random neighborhood structure.

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

BA

Initialisation:
Set maximum number of iteration, *NumOfIte*;
Set the population size, *populationsize*;
Set *ne*: number of elite solutions;
Set *nre*: recruited bees for elite solutions;
Set *nb*: number of best solutions;
Set *nrb*: recruited bees for remaining best solutions;
Set *stlim*: limit of stagnation cycles for abandonment solutions;
Initialise the population; Calculate the initial fitness value, *f*(*Sol*);
Set best solution, *Solbest* ← *Sol*;
iteration ← 0;
Improvement:
do while (*iteration* < *NumOfIte*)
  Rank all solutions in the population based on fitness value.
  nbSet ← Select the top *nb* solutions from the population;
  neSet ← Select the top *ne* solutions from the *nbSet*;
   for *i*=1: *nrb*
    for *j*=1: *nb*
     $Sol^* \leftarrow neSet_j$;
     Apply a random neighbourhood structure on *Sol** ;
     Update the population by the improved solutions;
    end for
   end for
   for *j*=1: *nre*
    for *h*=1: *nrb*
     $Sol^{**} \leftarrow neSet_j$;
     Apply a random neighbourhood structure on *Sol*** ;
    end for
   end for
   Solbest ← best solution found so far;
   Apply a random neighbourhood structure on the remaining (*nb*–*ne*) solutions and update the population;
   iteration++;
end do

Figure 1. Pseudo-code for the BA

A. Neighbourhood structures

We will consider that route R has the customers *a*, *b*, *c* and *d* in sequence {*a*→*b*→*c*→*d*}. The neighbourhood structures are presented as follows:

NBS1: One shift in the same route, where *b* is moved to a position after all the other customers' positions; the new sequence is {*a*→*c*→*d*→*b*}.

NBS2: Two shifts in the same route, where both *a* and *b* are moved to a position after *c* and *d*; the new sequence is {*c*→*d*→*a*→*b*}.

NBS3: One swap in the same route, where *a* and *c* are exchanged; the new sequence is {c→b→*a*→*d*}.

NBS4: Two swaps in the same route, where a and b are exchanged with c and d, the new sequence is {c→d→a→b}.

IV. Results and comparison

Solomon in 1987 proposes 6 datasets for the VRPTW, which have been used by a number of heuristics. In this paper, these benchmark datasets are used to test the performance of the BA. The instances vary in terms of the number of vehicles used to serve the customers, travelling time of the vehicles, vehicle capacity, customer locations and service time (for unloading/loading). In other words, each customer has his own time window, demand, location, earliest arrival time and latest arrival time, and service time.

All instances in the benchmark datasets have 100 customers; the travelling time among customers is equivalent to the same Euclidean distance. The 56 instances are split into six groups based on the arrangement of the customers' locations and time windows. These groups are C1, C2, R1, R2, RC1, and RC2. In group C the customers are clustered, while in group R the customers are distributed remotely. Group RC mixes the customers distribution in the instances R and C.

The experimental results are based on 10 independent runs. Our proposed algorithm were implemented in Java programming language on Intel® Core™ i3 processors. The execution times were between 20 and 700 seconds based on the size of the tested dataset.

The settings of the parameters have been used in the BA are presented in Table 1. Note that these values obtained from five run.

TABLE I. THE SETTINGS OF THE PARAMETERS USED IN THE BA

BA	# of iterations	500
	population size	50
	ne: number of elite solutions	2
	nre: recruited bees for elite solutions	30
	nb: number of best solutions	4
	nrb: recruited bees for remaining best solutions	10
	stlim: limit of stagnation cycles for abandonment solutions	10

To compare the performance of the algorithm with the state-of-the-art approaches a single objective is used, the distance is considered as the main objective.

TABLE II. PERFORMANCE COMPARISON BETWEEN DIFFERENT APPROACHES AND THE BA FOR VRPTW

Problem Class	Yassen et al. (2015)	Yu et al. (2011)	Braysy and Gendreu (2005)	Schulze and Fahle (1999)	Potvin and Bengio (1996)	Ho et al. (2001)	Lau et al. (2003)	The BA
C1	838.47	843.55	**828.38**	828.94	838.00	833.32	832.13	835.36
C2	605.41	611.12	**589.86**	589.93	589.90	593.00	**589.86**	593.74
R1	1207.76	1241.24	1222.12	1268.42	1296.83	**1203.32**	1211.55	1228.85
R2	977.19	961.11	975.12	1055.90	1117.70	951.17	1001.12	**931.56**
RC1	**1381.96**	1419.14	1389.58	1396.07	1446.20	1382.06	1418.77	1462.27
RC2	**1099.12**	1119.24	1128.38	1308.31	1360.60	1132.79	1170.93	1185.23
Sum	6109.91	6195.4	6133.42	6447.57	6649.23	**6095.66**	6224.36	6200.41

The average results of the six categories are used to quantify the contribution of each strategy where the BA is compared with seven other meta-heuristic approaches in the papers proposed by Yassen et al. (2015), Yu et al. (2011), Bräysy and Gendreau (2005), Schulze and Fahle (1999), Potvin and Bengio(1996), Ho et al. (2001) and Lau et al. (2003). The comparison of the total distance of each approach for each category is shown in Table 2.

It can be observed that the BA can provide good performance comparing with other algorithms such as; it is achieve the best result in category R2 and acceptable results in other categories where we did not use any improvement algorithms.

V. Conclusion

In this paper, the Bee Algorithm as a natural honeybee algorithm used to tackle the vehicle routing problem with time windows. Overall experimental results indicates that BA is able to obtain comparable results in comparison to state-of-the-art approaches, as represented by reducing the distance travelled, which is the main objective of the VRPTW.

Acknowledgements

This work was supported by the Ministry of Higher Education, Malaysia (FRGS/1/2015/ICT02/UKM/01/2 and ERGS/1/2013/ICT07/UKM/02/05), and the Universiti Kebangsaan Malaysia (DIP-2012-15).

References

[1] Baykasoglu, A., Ozbakir, L., & Tapkan, P. (2007). Artificial bee colony algorithm and its application to generalized assignment problem. *Swarm Intelligence: Focus on Ant and particle swarm optimization*, 113-144.

[2] Bhagade, A. S., & Puranik, P. V. (2012). Artificial bee colony (ABC) algorithm for vehicle routing optimization problem. Int. J. Soft Comput. Eng, 2, 329-333.

[3] Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms.*Transportation science*, *39*(1), 104-118.

[4] Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation science*, *39*(1), 119-139.

[5] Desrochers, M., Lenstra, J. K., Savelsbergh, M. W. P., & Sourris, F. (1988). Vehicle routing with time windows: optimization and approximation. Vehicle routing: method and studies. Studies in management science and systems-volume 16.*Publication of: Dalctraf*.

[6] Ho, W.K., Ang, J.C. and Lim, A., 2001. A Hybrid Search Alogrithm For The Vehicle Routing Problem With Time Windows. International Journal on Artificial Intelligence Tools, 10(03): 431-449.

[7] Ioannou, G., Kritikos, M., & Prastacos, G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of the Operational Research* Society, 52(5), 523-537.

[8] Jawarneh, S., & Abdullah, S. (2015). Sequential insertion heuristic with adaptive bee colony optimisation algorithm for vehicle routing problem with time windows. *PloS one*, *10*(7), e0130224.

[9] Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, *4*(3), 66.

[10] Lau, H. C., Lim, Y. F., & Liu, Q. (2001). Diversification of neighborhood via constraint-based local search and its application to VRPTW. In *Proc. CP-AI-OR 2001 Workshop*.

[11] Lau, H.C., Sim, M. and Teo, K.M., 2003. Vehicle routing problem with time windows and a limited number of vehicles. European Journal of Operational Research, 148(3): 559-569.

[12] Pham, D.T. & Castellani, M. 2009. The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimisation Problems. Proc. ImechE, Part C, 223(12), 2919-2938.

[13] Pham, D.T. Afify, A.A. & Koc, E. 2007. Manufacturing cell formation using the Bees Algorithm. In Proceedings of the Third International Virtual Conference on Intelligent production machines and systems (IPROMS 2007), Whittles, Dunbeath, Scotland, 523-528.

[14] Pham, D.T. Ghanbarzadeh, A. Koc, E. Otri, S. Rahim S. & Zaidi, M. 2005. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK.

[15] Pham, D.T. Kog, E. Ghanbarzadeh, A. Otri, S. Rahim, S. & Zaidi, M. 2006a. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, IPROMS 2006 Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems, Oxford, Elsevier. pp. 454-459.

[16] Pham, D.T. Otr,i S. Ghanbarzadeh, A. & Kog E. 2006b. Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition,

[17] ICTTA'06 Information and Communication Technologies, 1624-1629.

[18] Pham, D.T. Soroka, A.J. Ghanbarzadeh, A. Kog, E. & Otri S., Packianather, M. 2006d. Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm, IEEE International Conference on Industrial Informatics, 8, 1346-1351.

[19] Potvin, J.Y. and Bengio, S., 1996. The vehicle routing problem with time windows part II: genetic search. INFORMS journal on Computing, 8(2): 165-172.

[20] Schulze, J. and Fahle, T., 1999. A parallel algorithm for the vehicle routing problem with time window constraints. Annals of Operations Research (86): 585-607.

[21] Shi, Y. J., Meng, F. W., & Shen, G. I. (2012). A Modified Artificial Bee Colony Algorithm for Vehicle Routing Problems with Time Windows.

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

[23] Solomon, M. M., & Desrosiers, J. (1988). Survey paper-time window constrained routing and scheduling problems. *Transportation science*, *22*(1), 1-13.

[24] Toth, P., & Vigo, D. (2002). An overview of vehicle routing problems. In *The vehicle routing problem* (pp. 1-26). Society for Industrial and Applied Mathematics.

[25] Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: problems, methods, and* applications (Vol. 18). Siam.

[26] Yassen, E.T., Ayob, M., Ahmad Nazri, M.Z. and Sabar, N.R., 2015. A hybrid meta-heuristic algorithm for vehicle routing problem with time windows.International Journal on Artificial Intelligence Tools, 24(06), 1550021.

[27] Yu, B., Yang, Z.Z. and Yao, B.Z., 2011. A hybrid algorithm for vehicle routing problem with time windows. Expert Systems with Applications, 38(1): 435-441.