SOLVING THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS BY BEE COLONY OPTIMIZATION METAHEURISTIC

Miloš Nikolić
University of Belgrade, Faculty of Transport and Traffic Engineering, m.nikolic@sf.bg.ac.rs
Dušan Teodorović*
University of Belgrade, Faculty of Transport and Traffic Engineering, dusan@sf.bg.ac.rs
Milica Šelmić
University of Belgrade, Faculty of Transport and Traffic Engineering, m.selmic@sf.bg.ac.rs

***Abstract**: The vehicle routing problem with time windows is studied in the paper. This problem has been solved by Bee Colony Optimization metaheuristic. Proposed algorithm has been tested on the set of well-known benchmark examples. Obtained results show that the presented algorithm can find high-quality solutions.*

***Keywords:** Vehicle Routing Problem, Time Windows, Bee Colony Optimization.*

* Corresponding author

1. INTRODUCTION

Different versions of vehicle routing and scheduling problems on the network appear in all fields of transportation and logistics. Well-organized vehicle routing and scheduling can significantly contribute towards a decrease in transportation costs and increase the quality of transportation and logistic services. Numerous papers have been published in world literature during the past five decades treating different aspects of the vehicle routing problems (VRP). Significant reviews are given in the following papers: [3], [10], [12], and [20]. According to the vehicle routing problem definition every node is described by a certain *demand* (the amount to be delivered to the node or the amount to be picked up from the node). Other known values include the *coordinates* of the depot and nodes, the *distance* between all pairs of nodes, and the *capacity* of the vehicles providing the service. Vehicles leave the depot, serve nodes in the network, and after completion of their routes, return to the depot. The classical vehicle routing problem consists of finding the set of routes that minimizes transport costs.

In this paper we study the Vehicle Routing Problem With Time Windows (VRPTW). In the case of VRPTW there is the additional restriction that a time window is associated with each node ([4], [5]).This problem occurs in daily logistics activities, so it is obvious importance of its solving.

Since Bee Colony Optimization (BCO) meta-heuristic has become very powerful tool for solving hard combinatorial optimization problems, the authors decide to use it as a tool for solving VRPTW. BCO is a stochastic, random-search technique that belongs to the class of population-based algorithms. This technique uses an analogy between the way in which bees in nature search for food, and the way in which optimization algorithms search for an optimum of (given) combinatorial optimization problems.

The paper is organized as follows. Section after Introduction provides brief description of Bee Colony Optimization algorithm. Content of the following Section presents implementation of the BCO for VRPTW. Section named Numerical results contains experimental evaluations and the last Section brings conclusions related to the research.

2. BEE COLONY OPTIMIZATION

The Bee Colony Optimization metaheuristic has been introduced in the papers: [13], [14], [15] and [16]. Up to now BCO has been successfully applied to various engineering and management problems.

The basic idea behind BCO is to build the multi agent system (colony of artificial bees) that will search for good solutions of various combinatorial optimization problems exploring the principles used by honey bees during nectar collection process. Artificial bee colony usually consists of small number of individuals, but nevertheless, BCO principles are gathered from the natural systems. Artificial bees investigate through the search space looking for the feasible solutions. In order to find the best possible solutions, autonomous artificial bees collaborate and exchange information. Using collective knowledge and information sharing, artificial bees concentrate on the more promising areas, and slowly abandon solutions from the less promising ones. Step by step, artificial bees collectively generate and/or improve their solutions. The BCO search is running in iterations until some predefined stopping criterion is satisfied.

In the BCO algorithm every artificial bee generates one solution to the problem. Artificial bees generate/improve solutions in iterations. Each iteration consists of a certain number of *forward* and *backward* passes (Figure 1). In each forward pass, every artificial bee explores the search space. It applies a predefined number of moves, which construct and/or improve the solution, yielding to a new solution. Having obtained new solutions, the bees go again to the nest and start the second phase, the so-called backward pass. In the backward pass, all artificial bees share information about their solutions, i.e. they make known the quality of the generated solution (the objective function value). Through the backward pass, every bee decides with a certain probability whether to abandon the created solution and become uncommitted follower, or dance and thus recruit the nestmates before returning to the created solution (bees with higher objective function value have greater chance to continue its own exploration). Every follower, choose a new solution from recruiters by the roulette wheel (better solutions have higher probability of being chosen for exploration). The two phases of the search algorithm, forward and backward pass, are performed iteratively, until a stopping condition is met.

The BCO algorithm underwent numerous changes trough the process of evolution from its development, in 2001, until nowadays. Moreover, in order to solve hard combinatorial problems, the initial constructive BCO ([7], [8], [9], [13], [14], [15], [16]) was modified and a new concept based on the improving complete solutions ([6], [18], [19], [20]). was developed. Due to the nature of VRPTW problem, the authors decide to apply improvement version of BCO algorithm. The inputs of the BCO algorithm are:

B – the number of artificial bees,

IT – the number of iteration,

NP – the number of the forward and backward passes in a single iteration,

NC – the number of changes in one forward pass.

The output of the algorithm is:

S – the best obtained solution.

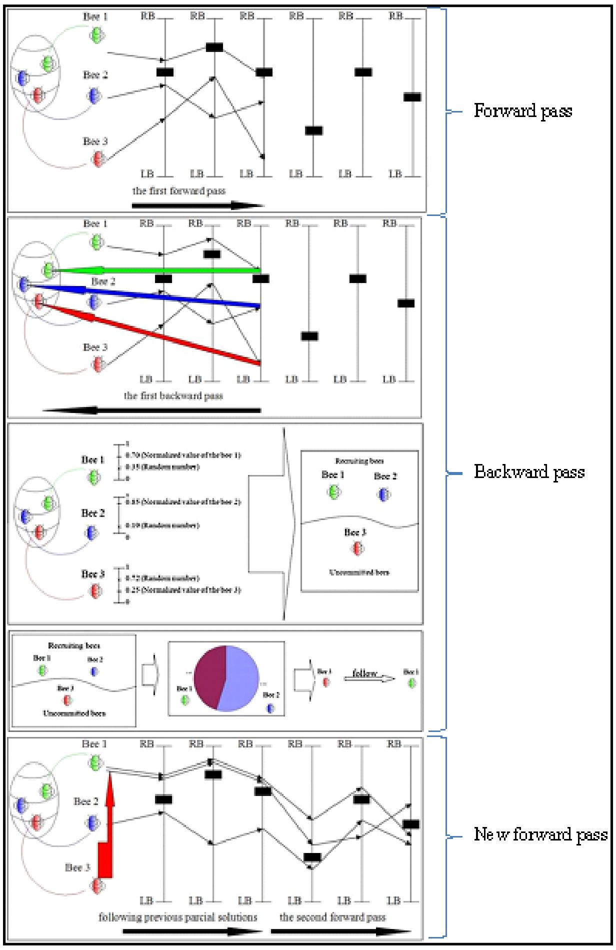

Figure 1. Forward and backward pass

The pseudo code of the BCO algorithm could be described in the following way:

***procedure BCO(in** B, IT, NP, NC, **out** S)*
Determine the initial solution.
S ← set the initial solution.
for** j = 1 **to** IT **do
for** i = 1 **to** B **do
the bee i ← the best solution S.
for** k = 1 **to** NP **do
for** i = 1 **to** B **do
for** r = 1 **to** NC **do
Evaluate possible changes in the solution of the bee i.
Taking into account evaluated values choose one change.
if** the best solution generated by the bees is **better** than the solution S **then
S ← the best bee's solution.
For** i = 1 **to** B **do
Evaluate solution of the bee i.

For** i = 1 **to** B **do
Make a decision whether the bee i is loyal.
For** i = 1 **to** B **do
if** the bee i is not loyal **then
Chose one of the loyal bees that will be followed by the bee i

3. THE BCO APPROACH TO THE VRPTW

In this paper, we propose the BCO heuristic algorithm tailored for the VRPTW. We propose the BCO algorithm that is based on the improvement concept. In other words, we first generate the initial feasible solution (the initial set of routes). Then, artificial bees investigate solution space in the neighborhood of the current solution, and try to improve the solution. The modification of solution is performed through *NP* forward passes with in the single iteration. We assume that at the beginning of a route design, all bees are in the artificial location-hive.

3.1 The initial solution generation

We use simple insertion heuristic to generate the initial solution. Our heuristic represents modification of the heuristic proposed in [2]. We simultaneously determine insertion location and the node that should be inserted in the route. We determine the first node in the route in a random manner. After that, all other nodes are inserted in the route according to insertion cost. We calculate the cost of insertion of node *u* between nodes *i* and *j* in the following way:

$$C_{iuj} = d_{iu} + d_{uj} - d_{ij} \tag{1}$$

where:
d_{ij} is the distance between nodes *i* and *j*.

The node with the lowest cost is inserted between the corresponding two nodes. If neither of unrouted nodes can be inserted in the route, the new route will be open, and the process will continue until the set of unrouted nodes is not empty.

3.2 Solution modification

We modify the solution in the way that couple of routes are removed and the nodes in these routes became unrouted. After that, we use insertion heuristic to make the new routes.

The paper [2] presented three insertion heuristics. The diference between them are in a way they calculate costs. Following the idea of these three insertion heuristics, we use three types of artificial bees.

The bees of the first type calculate cost as:

$$C^1_{iuj} = d_{iu} + d_{uj} - w\, d_{ij} \tag{2}$$

where:
w is the parameter (The larger w, the more emphasis is put on the distance between the vertices to be connected.).

The second type of bees calculate cost as:

$$C^2_{iuj} = w_1\, C^1_{iuj} + w_2\, l_s\big(t(j/u) - t(j)\big) \tag{3}$$

where:
w_1 and w_2 are the weighting coeficients $(w_1 + w_2 = 1)$,
l_s – is the parameter that depends on the insertion location: $l_s = 1 - \frac{s}{n+1}$, where *s* is the sequential number in the route where node *u* should be inserted and *n* is the total number of nodes in the route.
t(*j*) – arrival time at node *j* before inserting customer *u* in the route between *i* and *j*,
t(*j*/*u*) – arrival time at node *j* after inserting customer *u* in the route between *i* and *j*.

The bees of the third type calculate cost as:

$$C^3_{iuj} = w_1\, C^1_{iuj} + w_2\,\big(t(j/u) - t(j)\big) + w3\,(f(u) - t(u)) \tag{4}$$

where:
f(*u*) – beginning of the servce time at node *u*,
w_1, w_2 and w_3 are weighting coeficients $(w_1 + w_2 + w_3 = 1)$

3.3 Backward pass

All bees return to the hive after generating the solutions. All these solutions are then evaluated by all bees. (The total length of all routes characterizes every generated solution). We denote by O_i normalized value of the total length of all routes generated by the bee *i* :

$$O_i = \frac{T_{\max} - T_i}{T_{\max} - T_{\min}}, \tag{5}$$

where:
T_i – the objective function value (the total length of all routes) generated by the bee *b*,
T_{max}- the highest among all objective function values generated by all bees ($T_{\max} = \max_{i=1,n}\{T_i\}$),
T_{min}- the lowest among all objective function values generated by all bes ($T_{\min} = \min_{i=1,n}\{T_i\}$).

The probability that *i*-th bee (at the beginning of the new forward pass) is loyal to the previously discovered solution is calculated in the following way:

$$p_i = e^{-(O_{\max} - O_i)} \tag{6}$$

where:

O_{max} – maximum among all normalized values ($O_{max} = \max\limits_{i=1,n}\{O_i\}$)

If bee decides not to stay loyal to its previous solution, it should decide about the loyal bee she should follow. The probability that the bee will follow the loyal bee *i* can be calculated as:

$$p_i = \frac{O_i}{\sum\limits_{k \in L} O_k} \tag{7}$$

where:

L is a set of loyal bees.

Using equation (7) and a random number generator, each uncommitted follower joins one recruiter through a roulette wheel.

4. NUMERICAL RESULTS

The proposed algorithm is applied on a various test problems. In order to examine the effectiveness of BCO, the authors use Solomon's benchmark instances for the VRPTW. Parameters used for BCO implementation are the following:

- Number of iteration: *IT* = 1000,
- Number of passes: *NP* = 20,
- Number of changes in forward pass: *NC*=1
- Number of bees: *B* = 60 (20 of each type).

All instances are solved 10 times. The results are shown in the Table 1.

The first column of the Table 1 referes the name of the instance. The second column containes the best know result from literature [11]. Next two columns present the best results reached by the BCO algorithm, as well as relative error compared to the best known solution The last two columns show average solution obtained by 10 BCO runs and appropriate relative error.

As can be seen from Table 1. the BCO algorithm achieved results that differ from the best known in range from 0 to 6.19% depending on instance's difficultness. These results can be evaluated as satisfactory since this is the first attempt to apply BCO based on solution improvement to VRPTW. also, we didn't use any improvement algorithm such as 2-opt. all the tests were performed on Dell laptop with following characteristics: Intel core i5-2430m, 2.4 GHz, under Windows OS.

Table 1. Obtained results

Instance	Objective function value of the best known solution	BCO			
		The best obtained	Rel.Err. (%)	Average	Rel.Err (%)
R101	1642.88	1654.19	0.69	1664.21	1.30
R102	1472.62	1498.76	1.78	1517.32	3.04
R103	1213.62	1264.19	4.17	1293.96	6.62
R104	986.1	1006.2	2.04	1035.34	4.99
R105	1360.78	1389.02	2.08	1412.52	3.80
R106	1241.52	1281.17	3.19	1306.94	5.27
R107	1076.13	1119.7	4.05	1153.49	7.19
R108	948.57	975.922	2.88	1009.13	6.38
R109	1151.84	1199.35	4.12	1225.49	6.39
R110	1080.36	1147.27	6.19	1165.93	7.92
R111	1053.5	1112.42	5.59	1148.66	9.03
R112	953.63	987.804	3.58	1025.03	7.49
C101	828.94	828.937	0.00	828.94	0.00
C102	828.94	828.937	0.00	828.94	0.00
C103	828.06	852.19	2.91	866.53	4.65
C104	824.78	853.301	3.46	867.29	5.15
C105	828.94	828.937	0.00	828.94	0.00
C106	828.94	828.937	0.00	828.94	0.00
C107	828.94	828.937	0.00	828.94	0.00
C108	828.94	828.937	0.00	828.94	0.00
C109	828.94	833.696	0.57	844.82	1.92
RC101	1623.58	1654.94	1.93	1676.97	3.29
RC102	1466.84	1496.06	1.99	1523.25	3.85
RC103	1261.67	1301.27	3.14	1343.44	6.48
RC104	1135.48	1152.61	1.51	1179.30	3.86
RC105	1518.6	1554.93	2.39	1589.37	4.66
RC106	1377.35	1406.62	2.13	1430.54	3.86
RC107	1212.83	1236.74	1.97	1292.58	6.58
RC108	1117.53	1144.17	2.38	1168.21	4.53
Average			2.23		4.08

5. CONCLUSION

The Bee Colony Optimization is meta-heuristic technique created by the analogy with foraging behavior of honeybees, realizing the concepts of collective intelligence. A population of artificial bees searches for the optimal solution with cooperation that enables more efficiency and allows bees to reach the goals they could not achieve individually.

In this paper the BCO heuristic algorithm is used to tackle the vehicle routing problem with time windows. Authors applied the improvement concept of BCO. The proposed BCO algorithm is tested on various benchmark examples. Based on the preliminary results, we can conclude that BCO is able to produce high quality solution for all tested well known benchmark examples.

The achieved results indicate that the development of new models based on swarm intelligence principles could considerably contribute to the solution of difficult logistic problems.

ACKNOWLEDGMENT

This research was partially supported by Ministry of Education, Science and Technological Development Republic of Serbia, through the project TR36002 for the period 2011-2014.

REFERENCES

[1] Alvarenga, G. B., Mateus, G. R., de Tomi, G., 2007. *A genetic and set partitioning two-phase approach for the vehicle routing problem with time wondows*, Computers & Operations Research, 34, 1564 – 1584.

[2] Baker, E. K., Schaffer, J. R., 1986. *Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints*, Amer. J. Math. Management Sci., 6, 261-300.

[3] Bodin, L., Golden, B., Assad, A. A., Ball, M. O., 1983. *Routing and scheduling of vehicles and crews. The state of the art*, Computers and Operations Research, 10, 69-108.

[4] Bräysy O, Gendreau M., 2005. *Vehicle routing problem with time windows, Part I: Route Construction and Local Search Algorithms*, Transportation Science, 39, 1, 104-118.

[5] Bräysy O, Gendreau M., 2005. *Vehicle routing problem with time windows, Part II: Metaheuristics*, Transportation Science, 39, 1, 119-139.

[6] Davidović, T., Ramljak, D., Šelmić, M., Teodorović, D., 2011. *Bee colony optimization for the p-center problem*. Computers & Operations Research, 38, 1367- 1376.

[7] Davidović, T., Šelmić, M., Teodorović D., 2009. Scheduling Independent Tasks: Bee Colony Optimization Approach. Proceedings of the The 17th Mediterranean Conference on Control and Automation, MED'09, 1020-1025, Thessaloniki, Greece, June 24-26.

[8] Davidović, T., Šelmić, M., Teodorović, D., Ramljak, D., 2012. *Bee Colony Optimization for Scheduling Independent Tasks to Identical Processors*. Journal of Heuristics, 18,4, 549-569.

[9] Dimitrijević, B., Teodorović, D., Simić, V., Šelmić, M., 2011. *A Bee Colony Optimization Approach to Solving the Anti-Covering Location Problem*. Journal of Computing in Civil Engineering, 26, 6, 759-768.

[10] Golden, B. L., Assad, A. A. (Editors), 1998. *Vehicle Routing: Methods and Studies*, North-Holland, Amsterdam.

[11] Hamberto, C. B. O., Germano, C. V., 2010. *A hybrid search method for the vehicle routing problem with time windows,* Annals of Operations Research, 180, 125-144.

[12] Larson R. C., Odoni, A. R., 1981. *Urban Operations Research*, Prentice Hall, Englewood Cliffs, NJ.

[13] Lučić, P., Teodorović, D., 2001. *Bee system: Modeling combinatorial optimization transportation engineering problems by swarm intelligence*. In Preprints of the TRISTAN IV triennial symposium on transportation analysis, Sao Miguel, Azores Islands, Portugal (pp. 441–445).

[14] Lučić, P., Teodorović, D., 2002. *Transportation modeling: An artificial life approach*. In Proceedings of the 14th IEEE ''International conference on tools with artificial intelligence'', Washington, DC, 216–223.

[15] Lučić, P., Teodorović, D., 2003a. *Computing with bees: attacking complex transportation engineering problems*. International Journal on Artificial Intelligence Tools, 12, 375–394.

[16] Lučić, P., Teodorović, D., 2003b. *Vehicle routing problem with uncertain demand at nodes: The bee system and fuzzy logic approach*. In J. L. V erdegay (Ed.), Fuzzy sets in optimization, 67–82. Heidelberg, Berlin: Springer-Verlag

[17] Tan, K.C., Lee, L. H., Zhu, Q. L., Ou, K., 2001. *Heuristic method for vehicle routing problem with time windows, Artificial Intelligence in Engineering*, 15(1), 281-295.

[18] Nikolić, M., Teodorović, D., 2013. *Empirical study of the bee colony optimization (BCO) algorithm*. Expert Systems with Applications, 40(11), 4609–4620.

[19] Nikolić M., Teodorović D., 2013. *Transit network design by Bee Colony Optimization*. Expert Systems with Applications, 40, 15, 5945-5955.

[20] Solomon, M. M., Desrosiers, J., 1988. *Time Window Constrained Routing and Scheduling Problems*, Transportation Science, 22, 1-13.

[21] Todorović, N., Petrović, S., 2013. *Bee Colony Optimization for Nurse Rostering*, IEEE Transactions on Systems Man & Cybernetics: Systems, 43, 2, 467 - 473.