SOLVING THE VEHICLE ROUTING PROBLEM WITH HETEROGENEOUS VEHICLES BY THE BEE COLONY OPZIMIZATION METAHEURISTIC

Miloš Nikolić[a,*] , Milica Šelmić[a], Predrag Grozdanović[a]

[a] University of Belgrade – Faculty of Transport and Traffic Engineering

Abstract: In this paper, we consider the vehicle routing problem with heterogeneous vehicles. For this problem, we propose the Bee Colony Optimization metaheuristic. We tested our method on one benchmark example. Obtained results show that the metaheuristic significantly outperforms previous solutions given in the literature.

Keywords: *Vehicle routing problem, heterogeneous vehicles, Bee Colony Optimization*

1. INTRODUCTION

The vehicle routing problem (VRP) is one of the most studied combinatorial optimization problems. The main goal is to find the best vehicle routes in the network while minimizing one or more criteria. This problem was set up for a first time by Dantzig and Ramser (1959), while the first algorithm for solving the routing problems was posted by Clarke and Wright (1964). Till today many papers have been written dealing with the vehicle routing problem. During the time, the VRP has received numerous extensions. Some of these modifications are: the VRP with time windows (VRPTW), the VRP with simultaneous pickup and delivery (VRPSPD), the electric vehicle routing problem (E-VRP), etc.

This paper is considering a variant of the VRP problem which refers to routing the vehicles with different load capacity and volume. This variant is known as the vehicle routing problem with heterogeneous vehicles or as the heterogeneous fleet vehicle routing problem (HFVRP). The HFVRP problem was first defined and solved in the paper Golden et al. (1984). Li et al. (2007) developed a new variant of algorithm called record-to-record travel algorithm. This algorithm solves the standard VRP that considers a heterogeneous fleet. After Golden et al. (1984), the mentioned problem was solved by applying Tabu search metaheuristic in the papers Semet and Taillard (1993),Rochat and Semet (1994), Brandao (2011), etc. Prins (2002) developed a heuristic algorithm for solving the HFVRP problem. Primary goal of this algorithm was minimizing costs and the secondary goal was

* m.nikolic@sf.bg.ac.rs

the minimizing of number of vehicles used. This heuristic can easily face with various limitations and provides good starting solutions for using Tabu search metaheuristic.

Belmecheri et al. (2009) solved the problem called Heterogeneous Fleet VRPMB with time windows. They suggested the formula of integer linear programming and the Ant colony optimization algorithm (ACO). In the paper Ky Phuc and Phuong Thao (2021) the Ant colony optimization algorithm was proposed for solving the problem of routing multiple trucks and multiple delivery vehicles with a time frame and heterogeneous fleets. The problem of profitable heterogeneous vehicle routing problem with cross-docking was solving in the paper Baniamerian et al. (2019). They proposed a new hybrid metaheuristic algorithm based on a modified Variable neighborhood search with four shaking and two neighborhood structures and a Genetic algorithm was introduced to solve major problems.

By reviewing the literature, it can be concluded that the Bee Colony Optimization (BCO) metaheuristics has not been used to solve the VRP with heterogeneous fleet. So, this paper presents for the first time BCO metaheuristics for solving mentioned problem, the algorithm was tested and the obtained results are compared with the results from the paper of Sanz and Gomez (2013) who used Tabu search metaheuristic.

2. PROBLEM DESCRIPTION

We consider the vehicle routing problem where the customers have demands that are characterized by weight and volume. A company's fleet consists of the heterogeneous vehicles. Each vehicle has weight and volume capacities, and these capacities cannot be exceeded. There is just one depot at the transport network. All routes have to begin and end in that depot. Each customer should be served by just one vehicle. The aim is to find a set of routes in the way to minimize the total cost.

In this paper we calculate the total cost in the objective function in the same way as Sanz and Gomez (2013):

$$F = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} c_{ij}\, x_{ijk} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} ev_k\, d_{ij} x_{ijk} \quad (1)$$

where are:

n – the total number of nodes (the depot is the node 0 and the customers are the nodes from 1 to n),

c_{ij}- the cost for traveling from the node (customer or depot) i to the node j,

x_{ijk}- the binary decision that takes 1 if vehicle k after node i goes to the node j, and 0 otherwise,

d_{ij}- the distance from the node i to the node j,

ev_k – cost efficiency coefficient of the vehicle k.

3. BEE COLONY OPTIMIZATION

Bee Colony Optimization (BCO) is a metaheuristic proposed by Lučić and Teodorović (2001). This algorithm is based on the way how bees collect the nectar in the nature. The artificial bees try in a similar way to search the solution space of the considered combinatorial optimization problem. Up to now, this approach was used as the solution technique for many problems. Teodorović et al. (2021) gave a broad survey of the BCO applications in transport and traffic engineering.

There are two versions of this algorithm: the constructive (BCOc) and improvement version (BCOi). In the constructive version of the algorithm bees in each iteration start with the empty solution. During the iteration bees build their solutions. In the improvement version of the algorithm at the beginning of iterations bees have the same complete solution. Within iteration the bees try to improve that solution making small modifications.

In this paper, we have used the constructive version of the algorithm. The pseudo-code of this version of the algorithm can be given in the following way:

```
1.  do
2.      An empty solution is assigned to each bee.
3.      fori = 1 to the number of passes
4.          forj = 1 to the number of moves
5.              forb = 1 to the number of bees
6.                  Evaluate all possible moves of the bee b.
7.                  Choose one move of the bee b using the roulette wheel.
8.              nextb
9.          nextj
10.         forb = 1 to the number of bees
11.             Evaluate partial solution of the bee b.
12.         nextb
13.         Normalize quality of generated partial solutions.
14.         For each bee determine will it stay loyal to generated
            solution or not.
15.         forb = 1 to the number of bees
16.             If the bee b is not loyal to its solution, determine one loyal
                bee that the bee b will follow.
17.         nextb
18.     nexti
19.     Evaluate solutions of all bees.
20. while stopping criteria is not satisfied
21. Select the best generated solution.
```

An iteration of the algorithm consists of the steps 2 - 20 of the pseudo-code. Mainly the stopping criteria of the algorithm are the number of iterations and CPU time. The parameters of the BCO algorithm are: B - the number of the bees, NP – the number of passes and NM – the number of moves.

Steps 4 - 9 of the pseudo-code are known as forward pass, while steps 10 - 17 are the backward pass. Within the forward pass bees build their solutions, while in the backward pass bees compare their solutions and make loyalty decisions. When the bee is loyal to its solution, it continues to build that solution. Otherwise, if the bee is not loyal to its solution, it will select one of the loyal bees in order to follow it. That means, the bee, which is not loyal to its solution, will accept partial solution of the selected bee (which is loyal to its solution). Generally, in that way the bees with poor solutions most probably will decide to follow the other bees with better solutions.

In the considered problem we define that the number of moves in one forward pass should be determined at the beginning of each iteration. We define that the number of moves (NM) should be selected in the random manner between 1 and 5 ($NM = random[1,5]$).

Taking into consideration the total number of customers/nodes (let us denote this with n) and the number of move (NM) we can calculate the number of forward and backward passes (NP) in the following way:

$$NP = \lceil \frac{n}{NM} \rceil \quad (2)$$

In each forward pass, from 1st to $(NP-1)^{\text{th}}$, the bees have to make NM moves, and in the last forward pass they have to make $n - NM \cdot (NP - 1)$ moves.

In one move a bee must assign one customer to the one of the available vehicles. The customer could be assigned to the vehicle so as not to exceed the capacity constraints. When the customer is assigned to the vehicle, it will be put at the end of the vehicle's route. For example, if the vehicle has the current route (depot - 3 - 5 - 2), and the customer 7 is assigned to it, then the new route will be (depot - 3 - 5 - 2 - 7).

The customer j will be assigned to the vehicle k with the probability:

$$p_{jk} = \frac{\frac{1}{ev_k \, d_{ln(k),j}}}{\sum_{i \in N_b} \sum_{s \in K_i} \frac{1}{ev_s \, d_{ln(s),i}}} \quad (3)$$

where are:

ev_k – the efficiency coefficient of the vehicle k,

$ln(k)$ – the last node in the route of the vehicle k,

$d_{ln(k),j}$ – the distance between the last node (customer) in the route of the vehicle k and the customer j,

N_b – the set of customers (nodes) that is not yet assigned to any vehicles,

K_i – the set of vehicles to which customer i could be assign (taking into consideration vehicle capacity constraint)

ev_s – the efficiency coefficient of the vehicle s,

$ln(s)$ – the last node in the route of the vehicle s,

$d_{ln(s),i}$ – the distance between the last node (customer) in the route of the vehicle s and the customer i.

When all bees perform a predefined number of the forward passes and moves, all customers should be assigned to the vehicles. Taking into consideration that all vehicles have to return at the depot, when they finish last delivery, we have to add the depot at the end of each vehicle route.

In the considered problem we have to minimize the objective function (1). According to that, in the 13th step of the backward pass, we normalize the quality of partial solution of the bee b in the following way:

$$O_b = \frac{F_{max} - F_b}{F_{max} - F_{min}} \quad (4)$$

where are:

F_b – a quality of the partial solution generated by the bee b,

F_{max} – a maximal value of all F_b : $F_{max} = \max_{b=1,\cdots,B} \{F_b\}$,

F_{min} – a minimal value of all F_b: $F_{min} = \min_{b=1,\cdots,B} \{F_b\}$.

The probability that the bee b will stay loyal to its solution can be calculated in the following way:

$$p_b = e^{\frac{-(O_{max} - O_b)}{u}} \quad (5)$$

where are:

O_{max} - a maximal normalized value: $O_{max} = \max_{b=1,\cdots,B} \{O_b\}$

u – a current number of the forward pass.

Another very popular equation for the probability p_b is:

$$p_b = e^{-(O_{max} - O_b)} \quad (6)$$

To make a decision will bee b stay loyal to its solution, or not, we generate a random number,γ. The loyalty decision will be made in the following way: if $\gamma \leq p_b$ then the bee will stay loyal to its solution, otherwise it will not. The bees who's decided to stay loyal to its solutions are recruiters.

The bees who's decided not to stay loyal to its solution, have to make a decision to follow one of the recruiter bees. The probability that the bee, which decided not to stay loyal to its solution, will follow the recruiter bee r calculates in the following way:

$$p_r = \frac{O_r}{\sum_{k \in R} O_k} \quad (7)$$

where are:

O_r – a normalized value of the bee's r partial solution quality,

R – a set of the recruiter bees.

4. RESULTS

This chapter presents the comparison of the results of solving the HVRP given in Sanz and Gomez (2013) using Tabu search metaheuristic and using our BCO metaheuristic. A company that distributes foods in Querétaro, Mexico was used as study case. The company distributes a variety of foods in terms of weight and volume, and in addition, some types of food require a temperature regime. The company has a fleet of 3 vehicles, which differ in load capacity, volume and efficiency. The characteristics of each of the vehicles are shown in Table 1. Vehicle efficiency is a characteristic that can take values between 0 and 1, and is measured in relation to the energy consumption of that vehicle and maintenance costs. Vehicles with efficiency of 1 represent vehicles with ideal efficiency.

Table 1. Vehicle's efficiency, weight and volume capacity (Sanz and Gomez (2013))

Concept	Vehicle		
	1	2	3
Weights capacity (kg)	500	200	75
Volume capacity (m^3)	2.5	0.8	0.2
Efficiency	0.55	0.65	0.95

Table 2 shows the location of customers and their demands in terms of weight and volume of required food. The location of the clients is presented in the form of *x* and *y* coordinates, where it should be noted that all vehicles start from the depot with coordinates (0,0). We denote the depot as the node 0.

It should also be noted that the distance between two nodes is calculated using the Manhattan distance equation: $d(i,j) = |x_i - x_j| + |y_i - y_j|$.

Table 2. Customer's location, orders weight and volume (Sanz and Gomez (2013))

Customer *i*	x_i - coordinate	y_i - coordinate	Order weight [kg]	Order volume [m^3]
1	-2.1	-5.2	2.2	0.06
2	-3.1	-6.2	4.5	0.11
3	-8.8	-18.6	10.9	0.14
4	2.6	-25.2	12.2	0.19
5	0.8	-13.5	2.5	0.05
6	-1.1	-10.5	14.3	0.22
7	-1.2	-32.3	3.8	0.08
8	-8.5	-18.8	5.8	0.06
9	1.7	-21.2	4.4	0.09
10	0.5	-13.3	1.5	0.02
11	-3	-6.5	8.5	0.23
12	-0.9	-35.2	19.7	0.29
13	-8.3	-18.2	9.7	0.18
14	1.5	-20.7	4	0.19
15	10.3	-30.5	5.5	0.14
16	0.7	-14.6	6.2	0.12
17	-0.8	-30.2	4.9	0.6
18	-0.5	-12.8	12.5	0.29
19	2.8	-24.3	3.2	0.06
20	-0.3	-6.5	5.5	0.19

The existing routes used by the company for each vehicle in the fleet are generated using a typical assignment problem. These routes are shown in Table 3 as typical solution (Sanz and Gomez, 2013). The table also shows the set of routes obtained by Tabu search algorithm (Sanz and Gomez, 2013). The qualities of both solutions are given in the last column of the table.

Table 3. The vehicle routes proposed by Sanz and Gomez (2013)

Solution	Vehicl e	Routing assignment	Objective function
Typical	1	0-16-10-18-13-3-14-9-19-4-17-7-12-15-0	142.33
	2	0-11-2-20-6-0	
	3	0-1-5-8-0	
Tabu search	1	0-18-16-14-19-4-17-7-12-15-8-3-13-0	111.4
	2	0-9-5-10-6-11-2-1-0	
	3	0-20-0	

We applied the BCOc algorithm on the same example. The algorithm was implemented in Java programming language, using Appach NetBeans 13 editor. All tests were made in the desktop computer with the following performances: AMD Ryzen 7 3800 X with 32 GB of RAM memory, operating system: Ubuntu 21.10.

We made two tests of the BCOc algorithm. The difference between them is in the expression how the probability for the loyalty decision is calculated. In the first test we used the equation (5), and in the second test we used the equation (6).

In both tests we used the CPU time as the stopping criteria. We provided results for three different values of the CPU time (0.5, 1 and 2 minutes), and for three values of the number of bees (10, 15 and 20 bees). For each of these 9 options we did 5 runs of the algorithm. The obtained results are given in Tables 4 and 5.

In the first test the best solution has the objective function value 93.96. This solution was obtained with 15 bees and 1 minute of the allowed CPU time. The worst solution has the objective function value 98.27 (obtained with 15 bees and 0.5 minute of the CPU time). The best average objective function value, 95.19, was obtained with 10 bees and 2 minutes.

Table 4. The BCOc algorithm results when equation (5) is used

Run	10 bees			15 bees			20 beeswhen equation (5) is used		
	CPU time			CPU time			CPU time		
	0.5 min	1 min	2 min	0.5 min	1 min	2 min	0.5 min	1 min	2 min
1	97.65	95.94	94.4	96.66	95.61	96.38	96.61	94.64	96.95
2	97.1	97.22	95.48	98.27	95.61	94.73	95.83	96.88	94.88
3	95.15	96.6	95.43	97.15	97.82	96.42	94.62	96.9	95.19
4	95.81	95.72	94.86	97.69	97.08	96.95	98.16	97.15	96.18
5	94.73	97.22	95.78	96.22	93.96	96.03	94.99	96.31	96.16
Min	94.73	95.72	94.4	96.22	**93.96**	94.73	94.62	94.64	94.88
Max	97.65	97.22	95.78	**98.27**	97.82	96.95	98.16	97.15	96.95
Average	96.09	96.54	**95.19**	97.20	96.02	96.10	96.04	96.38	95.87

According to the results obtained with the BCOc algorithm when equation (6) is used, it can be noticed that the best solution has the objective function value 92.75. This solution was obtained for the two set of the parameters (20 bees and 1 minute of the CPU time, 20 bees and 2 minutes of the CPU time). The worst solution has the objective function value 95.54 and it was obtained for 20 bees and 0.5 minute of the CPU time. The best average objective function value, 93.34, was obtained for 10 bees and 2 minutes of the CPU time.

Table 5. The BCOc algorithm results when equation (6) is used

Run	10 bees			15 bees			20 bees		
	CPU time			CPU time			CPU time		
	0.5 min	1 min	2 min	0.5 min	1 min	2 min	0.5 min	1 min	2 min
1	94.75	92.97	93.41	94.75	94.99	93.41	95.54	94.71	93.3
2	94.69	93.69	93.47	93.85	93.36	93.14	94.53	93.74	93.74
3	93.41	93.3	93.41	94.25	94.88	94.18	93.36	94.07	93.3
4	94.69	93.69	93.47	93.69	93.8	92.97	94.73	92.75	92.75
5	94.77	94.31	92.92	94.82	93.96	93.3	94.44	93.85	93.96
Min	93.41	92.97	92.92	93.69	93.36	92.97	93.36	**92.75**	**92.75**
Max	94.77	94.31	93.47	94.82	94.99	94.18	**95.54**	94.71	93.96
Average	94.46	93.59	**93.34**	94.27	94.20	93.40	94.52	93.82	93.41

Among all generated solution the best one is with the objective function value 92.75. This solution is given in Table 6. The weight of the goods on the first route is 106.8 kg and the volume is 2.5 m^3. The second route has the total weight of the goods 29.5 kg and the volume 0.62 m^3. The third route has 5.5 kg and 0.19 m^3 of the goods. It can be noticed that this solution has much better objective function value than the best solutions given by Sanz and Gomez (2013). The relative differences are:

- from the typical solution: $\frac{\mathbf{142.33-92.75}}{\mathbf{142.33}} \cdot \mathbf{100}\ \% = \mathbf{34.83}\ \%$
- from the Tabu search solution: $\frac{\mathbf{111.4-92.75}}{\mathbf{111.4}} \cdot \mathbf{100}\ \% = \mathbf{16.74}\ \%$.

Table 6. The best obtained solution using BCOc

Route 1	0 – 10 – 5 – 16 – 13 – 3 – 8 – 14 – 9 – 19 – 4 – 15 – 7 – 12 – 17 – 18 – 0
Route 2	0 – 6 – 11 – 2 – 1 – 0
Route 3	0 – 20 – 0

5. CONCLUSION

The vehicle routing problem is very important transportation engineering problem. This problem belongs to the group of hard combinatorial optimization problems. To solve this problem researcher mainly use heuristic and metaheuristic algorithms.

In this paper we considered the vehicle routing problem with the heterogeneous vehicles. We applied the constructive version of the Bee Colony Optimization (BCOc) metaheuristic. The developed algorithm was tested on the example given by Sanz and Gomez (2013). This example is from the Querétaro, city in Mexico, and it includes 20 customers. The best obtained solution by the BCOc algorithm is significantly better than the solutions given by Sanz and Gomez (2013). In the future research, the proposed algorithmshould be tested on more examples and compared with other approaches.

REFERENCES

[1] Baniamerian, A., Bashiri, M., Tavakkoli-Moghaddam, R. (2019). Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. Applied Soft Computing Journal, 75, 441–460, 2019.

[2] Belmecheri, F., Prins, C., Yalaoui, F., Amodeo, L. (2009). An Ant Colony Optimization algorithm for a Vehicle Routing Problem with Heterogeneous fleet, Mixed Backhauls, and Time Windows. 13th IFAC Symposium on Information Control Problems in Manufacturing, 1550-1555.

[3] Brandao, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. Computers & Operations Research,38, 140–151.

[4] Clarke, G., Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations research, 12, 4.

[5] Dantzig, G.B., Ramser, J.H. (1959). The truck dispatching problem. Management Science, 6(1), 80–91.

[6] Golden, B.L., Assad, A.A., Levy L., Gheysens, F.G. (1984). The fleet size and mix vehicle routing problem. Computers & OR, 11(1), 49–66.

[7] Ky Phuc, P.N., Phuong Thao, N.L. (2021). Ant Colony Optimization for Multiple Pickup and Multiple Delivery Vehicle Routing Problem with Time Window and Heterogeneous Fleets. Logistics, 5, 28.

[8] Li, F., Golden, B., Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers & Operations Research, 34, 2734 – 2742.

[9] 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.

[10] Prins, C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. Journal of Mathematical Modelling and Algorithms, 135–150.

[11] Rochat, Y., Semet, F. (1994). A tabu search approach for delivering pet food and flour in Switzerland.Journal of the Operational Research Society, 45, 1233–1246.

[12] Sanz, F.T., Gomez, E.N.E. (2013). The vehicle routing problem with limited vehicle capacities. International Journal for Traffic and Transport Engineering, 3(3), 260 - 268.

[13] Semet, F., Taillard E. (1993). Solving real-life vehicle routing problems efficiently using tabu search. Annals of Operations Research, 41, 469–488.

[14] Teodorović, D., Davidović, T., Šelmić, M., Nikolić, M. (2021). Bee Colony Optimizaiton and Its Applications, In Handbook of AI-based Metaheuristics (Kulkarni, A.J. and Siarry, P. eds), 1st Edition, CRC Press, Boca Raton.