Artificial bee colony algorithm for the capacitated vehicle routing problem

1 author:

Ivona Brajevic
University Business Academy in Novi Sad

Artificial bee colony algorithm for the capacitated vehicle routing problem

Ivona BRAJEVIC
Faculty of Mathematics
University of Belgrade
Studentski trg 16
SERBIA
ivona.brajevic@googlemail.com

Abstract: - This paper presents an artificial bee colony (ABC) algorithm adjusted for the capacitated vehicle routing problem. The vehicle routing problem is an NP-hard problem and capacitated vehicle routing problem variant (CVRP) is considered here. The artificial bee metaheuristic was successfully used mostly on continuous unconstrained and constrained problems. Here this algorithm has been implemented for rather different type of problems and tested on twelve benchmark instances of small scale problems. Our results were compared to the best known results. The computational study showed that the proposed algorithm is a good and promising approach for capacitated vehicle routing problem.

Key-Words: - Artificial bee colony, Capacitated vehicle routing problem, Metaheuristic optimization, Swarm intelligence

1 Introduction

The vehicle routing problem (VRP) was first introduced by Dantzig and Ramser in 1959 and it has been widely studied since. This difficult combinatorial problem contains both, bin packing problem (BPP) and the travelling salesman problem (TSP) as special cases and conceptually lies at the intersection of these two well-studied problems [1]. Because of the interplay between the two underlying models, instances of the vehicle routing problem can be extremely difficult to solve in practice. Vehicle routing problem is NP-hard and finding an optimal solution to an NP-hard problem is usually impossible or very time consuming. Exact solution of the VRP thus presents an interesting challenge to which various approaches have been proposed. Because of this nature of the problem, it is not realistic to use exact methods to solve large instances of the problem. Most approaches are based on heuristics. Several, mostly hybrid, heuristics have been applied for solving the VRP problem. Some of them are: a hybrid search method which associates non-monotonic simulated annealing to hill-climbing and random restart [2], hybrid discrete particle swarm optimization algorithm (DPSO) [3], an improved ant colony optimization (IACO) [4], honey bees mating optimization algorithm [5], optimized crossover genetic algorithm [6].

During the last decade nature inspired intelligence has become increasingly popular. Among the most popular nature inspired metaheuristics are those methods representing successful animal and micro-organism team behavior, such as swarm or flocking intelligence or artificial immune systems. Swarm intelligence based on ant colonies was studied [7], [8], and recently also based on bees [9], [10], [11], [12], [13].

One approach that simulates the foraging behavior of the bees is the artificial bee colony (ABC) algorithm proposed by Karaboga [9]. The artificial bee colony algorithm is mainly applied to continuous optimization problems. Karaboga and Basturk have compared the performance of the ABC algorithm with the performance of other well-known modern heuristic algorithms. It has been shown that the ABC algorithm can be efficiently used for solving unconstrained and constrained optimization problems [10].

In this work, the ABC algorithm was applied to the capacitated vehicle routing problem (CVRP), one version of the VRP problem [14], [15]. In order to test the proposed algorithm we used twelve benchmark instances of small scale problems. This paper is organized as follows. Section 2 presents the vehicle routing problem. Section 3 describes the

This research is supported by Ministry of Science, Republic of Serbia, Project No. 44006

ABC algorithm. In Section 4 the implementation details of the ABC algorithm for the CVRP problem are described. Section 5 presents benchmark functions and test results. Conclusion and some possible plans for future work are in Section 6.

2 Problem description

Many versions of the vehicle routing problem have been described. The capacitated vehicle routing problem (CVRP) is discussed here and it can be described as follows:

Goods are to be delivered to a set of customers by a fleet of vehicles from a central depot. Each vehicle has limited capacity and each customer has a certain demand. The locations of the depot and the customers, the capacity of each route and the demand of each customer are given. The objective is to determine a viable route schedule which minimizes the distance or the total cost with the following constraints:

1. Each customer is served exactly once by exactly one vehicle
2. Each vehicle starts and ends its route at the depot
3. The total length of each route must not exceed the constraint
4. The total demand of any route must not exceed the capacity of the vehicle

Assume that the depot is node *0*, and *N* customers are to be served by *K* vehicles. The demand of customer *i* is q_i, the capacity of vehicle *k* is Q_k, and the maximum allowed travel distance by vehicle *k* is D_k. Then the mathematical model [16] of the CVRP is described as follows:

$$\text{Minimize } \sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij}^{k} X_{ij}^{k} \tag{1}$$

subject to:

$$X_{ij}^{k} = \begin{cases} 1, \text{ if vehicle k travel from custemer i to j} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

$$\sum_{k=1}^{K}\sum_{i=0}^{N} X_{ij}^{k} = 1, \quad j = 1,2,\ldots,N \tag{3}$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} X_{ij}^{k} = 1, \quad i = 1,2,\ldots,N \tag{4}$$

$$\sum_{i=0}^{N} X_{it}^{k} - \sum_{j=0}^{N} X_{tj}^{k} = 0, \; k = 1,2,\ldots,K; \; t = 1,2,\ldots,N \tag{5}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}^{k} X_{ij}^{k} \le D_k, \; k = 1,2,\ldots,K \tag{6}$$

$$\sum_{j=0}^{N} q_j \left(\sum_{i=0}^{N} X_{ij}^{k} \right) \le Q_k, \; k = 1,2,\ldots,K \tag{7}$$

$$\sum_{j=1}^{N} X_{0j}^{k} \le 1, \; k = 1,2,\ldots,K \tag{8}$$

$$\sum_{i=1}^{N} X_{i0}^{k} \le 1, \; k = 1,2,\ldots,K \tag{9}$$

$$X_{ij}^{k} \in \{0,1\}, \; i, j = 0,1,\ldots,N, \; k = 1,2,\ldots,K \tag{10}$$

where *N* represents the number of customers, *K* is the number of vehicles, C_{ij}^{k} is the cost of travelling from customer *i* to customer *j* by vehicle *k* and d_{ij}^{k} is the travel distance from customer *i* to customer *j* by vehicle *k*. The objective function Eq. (1) is to minimize the total cost by all vehicles. Constraints Eq. (3) and Eq. (4) ensure that each customer is served exactly once. Constraint Eq. (5) ensures the route continuity. Constraint Eq. (6) shows that the total length of each route has a limit. Constraint Eq. (7) shows that the total demand of any route must not exceed the capacity of the vehicle. Constraints Eq. (8) and Eq. (9) ensure that each vehicle is used no more than once. Constraint Eq. (10) ensures that the variable only takes the integer 0 or 1.

The vehicle routing problem is of great practical significance in real life. It appears in a large number of practical situations, such as transportation of people and products or delivery service.

3 Artificial bee colony algorithm

Artificial bee colony algorithm proposed by Karaboga [9] and later modified by Karaboga and Akay [11] is a new metaheuristic technique inspired by the foraging behavior of natural honey bee swarms. In ABC algorithm the colony of artificial bees consists of three groups of bees: employed bees, onlookers and scouts. All bees that are currently exploiting a food source are known as employed bees. The number of the employed bees is equal to the number of food sources. Onlookers are those bees that are waiting in the hive for the employed bees to share information about the food sources presently being exploited by them. Usually, the number of onlookers is taken to be equal to the number of employed bees. Employed bees share information about food sources by dancing in the dance area inside the hive. The dance is dependent on the nectar content of food source just exploited by the dancing bee. Onlooker bees watch the dance and choose a food source according to the probability proportional to the quality of that food source. Therefore, good food sources attract more onlooker bees compared to bad ones. Scouts are those bees that are searching for new food sources in the neighborhood of the hive. The employed bee whose food source has been abandoned by the bees becomes a scout. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees can be visualized as performing the job of exploitation. Each food source is a possible solution for the problem and the nectar amount of a food source represents the quality of the solution represented by the fitness value.

Short pseudo–code of the ABC algorithm is given below [11]:

```
Initialize the population of solutions
Evaluate the population
cycle = 1
repeat
  Employed bee phase
  Calculate probabilities for onlookers
  Onlooker bee phase
  Scout bee phase
  Memorize the best solution achieved so far
  cycle = cycle+1
until cycle = Maximum Cycle Number
```

4 ABC algorithm for vehicle routing problem

The ABC algorithm is usually used for continuous optimization problems. To apply it to integer (combinatorial) or mixed problems, modifications and adjustments are needed.

4.1 Solution encoding

To encode a solution [3] in which N customers are to be served by K vehicles we used a *2D* array. The first row in the *2D* array of the solution is $N \times K$ dimension vector $(s_1, s_2, ..., s_{N\times K})$, where s_i, $i=1,2,...\ N \times K$, is a natural number in the range [1, $N \times K$], which is not equal to each other. The number $l = s_i - \lfloor (s_i - 1)/N \rfloor \times N$ represents the l^{th} customer and the number $k = \lfloor (s_i - 1)/N \rfloor + 1$ represents the k^{th} vehicle. The second row in the *2D* array is also $N \times K$ dimension vector, where every position takes *0* or *1*. If the $(s_i)^{th}$ bit takes *1*, it represents that the l^{th} customer will be served by the k^{th} vehicle. Otherwise, the l^{th} customer will not be served by the k^{th} vehicle. Fig. 2 shows an example of solution encoding.

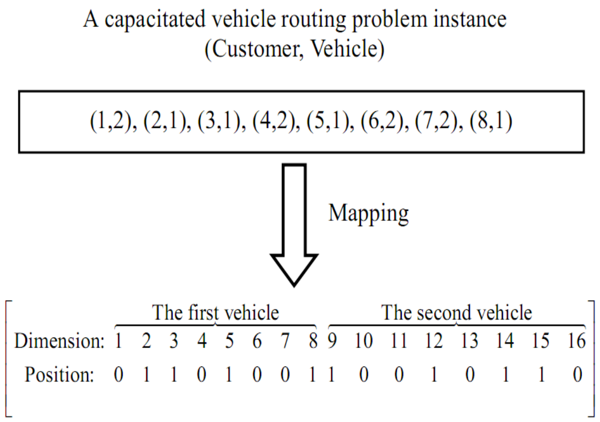

Fig.2 An example of encoding the solution of the CVRP instance with 8 customers and 2 vehicles

4.2 Generation of initial solutions and constraints handling

ABC algorithm generates randomly distributed initial solutions. In that way we cannot ensure the constraints are satisfied, so during initialization we must check every solution according to each route as follows:

- If the value of more than one position in the corresponding positions of all routes in the solution is 1, we randomly select one position on these positions and make its value be 1 and the others be 0.
- If the values of the corresponding positions of all routes in the solution are all 0, we randomly select one position and make its value be 1, and the others unchanged.

By this initialization we ensure for every initial solution that each customer is served exactly once by exactly one vehicle. If the total length in the route exceeds the limited value or the total demand of each route exceeds the capacity of the vehicle, the solution is infeasible.

In order to handle the constraints, the ABC algorithm employs Deb's rules [17], which are used in the version of the ABC proposed for constrained optimization problems. The method uses a tournament selection operator, where two solutions are compared at a time by applying the following criteria:

- Any feasible solution satisfying all constraints is preferred to any infeasible solution violating any of the constraints
- Among two feasible solutions, the one having better fitness value is preferred
- Among two infeasible solutions, the one having the smaller constraint violation is preferred

Because initialization with feasible solutions is a very time consuming process and in some cases it is impossible to produce a feasible solution randomly, the ABC algorithm does not consider the initial population to be feasible. Structure of the algorithm already directs the solutions to feasible region in running process due to the Deb's rules.

4.3 Neighborhood operators

In order to produce new solutions for the employed and onlooker bees we use two neighborhood operators [18]. The first neighborhood operator is called SwapMutation. It was introduced as a neighborhood operator for solving the travelling salesman problem (TSP) and is called "2-change". The idea of the mutation operation is to randomly mutate the tour and hence produce a new solution *g* that is not very far from the original one *f*. In this paper, the mutation operator is designed to conduct customer exchanges in a random fashion. The steps for the SwapMutation operator are as follows:

1. Randomly select two routes from the solution *f* and randomly select two customers from each selected route.
2. Exchange the customers in the different routes and generate the new solution *g*.

The second neighborhood operator, also based on random changes, is called InsertMutation. The steps for the InsertMutation operator are as follows:

1. Randomly select the routes from the solution *f* and randomly select one customer from one selected route.
2. Remove the selected customer from one route to the other and generate the new solution g.

After the new solution is produced we apply selection process based on Deb's method. If the new solution *g* is accepted instead of the solution *f* , we apply SwapMutation operator in the mutated routes in order to improve the new solution. SwapMutation is applied in the following way:

1. Compare all possible pairwise exchanges of customer locations in the route to find the exchange that produces the shortest distance.
2. When the pair of customers whose exchange produces the shortest distance is found, the route is rearranged. If we don't find such pair of customers, the route stays unchanged.

In employed bee phase SwapMutation operator is applied two times. Onlooker bee phase has one difference from employed bee phase: it applies one SwapMutation and one InsertMutation operator instead of two SwapMutation operators.

4.4 ABC adjustment

The pseudo–code of the proposed ABC algorithm is given below:

```
Construct initial employed bee colony solutions;
Evaluate fitness value for each solution;
cycle = 1;
repeat
  For each employed bee apply two times:
    - SwapMutation operator
    - Selection process based on Deb's method
    - If the new solution is accepted, improve the
      new solution and evaluate them
  Calculate the probability values for the solutions
  For the onlookers selected depending on
  probabilities, the following is applied two times:
    - First time Swap Mutation operator,
      second time Insert Mutation operator
    - Selection process based on Deb's method
    - If the new solution is accepted, improve
      The new solution and evaluate them
  Determine every infeasible solution for the
   scout and replace them with new produced
   solutions
  Memorize the best solution achieved so far
  cycle = cycle + 1
until cycle = Maximum Cycle Number
```

The expressions for evaluating fitness value for the population and the probabilities for onlookers stayed the same as in the version of the ABC proposed for unconstrained optimization problems. In the initialization phase only the first initialization of food sources is completely random. In other initialization phases the first new food source is the food source from the previous run of the algorithm which has the best fitness value. Therefore, exploitation of the good sources was increased. In order to increase the exploration the scout bee's phase was changed. In the scout phase the algorithm

ISBN: 978-960-474-297-4

checks every possible solution. If the solution is not feasible, that food source is replaced with a new randomly produced solution. The number of scouts in the particular iteration is changeable, it depends on the number of infeasible solutions.

5 Test and results

The ABC algorithm is applied to the twelve small problem instances which have been widely used as benchmarks in order to compare its ability to find the solution to VRP. The problem instances belong to different groups of known benchmark problems. Instances gr-n17-k3, gr-n21-k3 and fri-n26-k3 are converted TSPLIB problems. B-n31-k5, B-n35-k5, B-n38-k6, B-n43-k6, P-n16-k8, P-n19-k2 and P-n21-k2 belong to the problem sets B and P of Augerat. Problems E-n13-k4 and E-n23-k3 belong to the Christofides and Elion problems. All these problems are from [19]. Tested instances and their characteristics - number of customers (N), number of vehicles (K), tightness (Demand/Capacity) and the best known solutions (BKS) are given in Table 1

Problem	N	K	Tightness	BKS
gr-n17-k3	16	3	0.89	2685
gr-n21-k3	20	3	0.95	3704
fri-n26-k3	25	3	0.83	1353
B-n31-k5	30	5	0.82	672
B-n35-k5	34	5	0.87	955
B-n38-k6	37	6	0.85	805
B-n43-k6	42	6	0.87	742
P-n16-k8	15	8	0.88	450
P-n19-k2	18	2	0.97	212
P-n21-k2	20	2	0.93	211
E-n13-k4	12	4	0.76	247
E-n23-k3	22	3	0.75	569

Table 1. List of tested instances

Control parameters of the ABC algorithm are: swarm size, limit, number of employed bees, number of onlookers, number of scouts and maximum number of cycles. In these experiments, the colony size was 50 and the maximum number of cycles was 150 for the instances up to 25 nodes, and 500 for the instances up to 38 nodes. Each experiment was repeated 15 runs. For instance B-n43-k6 the maximum number of cycles was 2000 and each experiment was repeated 5 runs. The percentages of onlooker bees and employed bees were 50% of the colony and the number of scout bees was changeable, for every iteration it was equal to the number of current infeasible solutions. The value of "limit" was not important, because any feasible solution was not abandoned during one run of the algorithm. The performance of the algorithm was considered in terms of the best and mean values, and its performance was compared with the best known optimal solutions. ABC algorithm has been implemented in Java programming language and run on a Pentium Core2Duo, 1.40-GHz personal computer with 2 GB RAM memory.

Parameters adopted for ABC algorithm are given in Table 2.

Control parameters for ABC algorithm	
swarm size	50
number of employed bees	50% of the swarm
number of onlookers	50% of the swarm
number of scouts	changeable

Table 2. Control parameters adopted for the ABC algorithm

Table 3 shows the best and mean results obtained by the ABC algorithm for each of the instance tested.

Problem	BKS	Best	Mean
gr-n17-k3	2685	2685	2685.7
gr-n21-k3	3704	3704	3810.6
fri-n26-k3	1353	1358	1366.6
B-n31-k5	672	672	675.3
B-n35-k5	955	990	999.1
B-n38-k6	805	820	828.8
B-n43-k6	742	772	801,4
P-n16-k8	450	450	451.7
P-n19-k2	212	212	212.9
P-n21-k2	211	211	216.1
E-n13-k4	247	247	247.0
E-n23-k3	569	569	595.5

Table 3. Computational results obtained by ABC algorithm

The results from Table 3 show that the best obtained results by the ABC algorithm are close to the best known solutions, except for the instance B-n35-k5 and instance B-n43-k6 where the percentage difference from the optimum value is about 4%. It can be seen that the algorithm can converge quickly towards the global optimum for most of the tested instances (gr-n17-k3, fri-n26-k3, E-n13-k4, P-n16-k8, B-n31-k5). For the other tested benchmarks the convergence is slower and the algorithm may need to be modified to avoid being trapped at some local attractors.

ISBN: 978-960-474-297-4

6 Conclusion

In this paper, we presented the ABC algorithm for capacitated vehicle routing problem. The twelve benchmark instances of small scale problems were tested. The results were compared to the best known results. Although the global optimality cannot be guaranteed, the performance of the algorithm is good and robust. It is noticed that algorithm can be trapped in the local minimum for some benchmark instances. In the future work the algorithm needs to be explored and tested for larger instances of the CVRP. The proposed approach is also suitable for other combinatorial problems.

References:

[1] Ralphs T. K.,Kopman L., Pulleyblank W.R., L.E. Trotter, On The Capacitated Vehicle Routing Problem, *Mathematical Programming* (2003) Vol. 94, Issue 2, pp. 343-359

[2] Oliveira H. C. B., Vasconcelos G. C., A hybrid search method for the vehicle routing problem, *Annals of Operations Research*, Vol. 180, No. 1, 2010, pp. 125-144

[3] Chen A., Yang G., Wu Z., Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, *Journal of Zheijang University SCIENCE A*, Vol. 7, No. 4, 2006, pp. 607-614

[4] Yu B., Yang Z., Yao B., An improved ant colony optimization for vehicle routing problem, *European Journal of Operational Research*, Vol. 196, Issue 1, 2009, pp. 171–176

[5] Marinakis Y., Marinaki M., Dounias G., Honey Bees Mating Optimization algorithm for large scale vehicle routing problems, *Natural Computing*, Vol. 9, Issue 1, 2010, pp. 5-27

[6] Nazif H., Lee L. S., Optimized Crossover Genetic Algorithm for Vehicle Routing Problem with Time Windows, *American Journal of Applied Sciences*, Vol. 7, Issue 1 2010, pp. 95-101

[7] Dorigo M, Gambardella LM: Ant colonies for the traveling salesman problem, *BioSystems* Vol. 43, No.2 ,1997, pp.73-81.

[8] Tuba M., Jovanovic R., An Analysis of Different Variations of Ant Colony Optimization to the Minimum Weight Vertex Cover Problem, *WSEAS Transactions on Information Science and Applications*, Vol. 6, Issue 6, June 2009, pp. 936-945

[9] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, *Technical Report-TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

[10] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) Algorithm, *Applied Soft Computing*, Vol. 8, Issue 1, 2008, pp. 687-697

[11] D. Karaboga, B. Akay, A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems, *Applied Soft Computing*, Article in print, 2010, doi:10.1016/j.asoc.2010.12.001

[12] Bacanin N., Tuba M., Brajevic I., An Object-Oriented Software Implementation of a Modified Artificial Bee Colony (ABC) Algorithm, *Recent advances in neural networks, fuzzy systems & evolutionary computing*, WSEAS Press, June 2010, pp. 179-184

[13] Subotic M., Tuba M., Stanarevic N., Parallelization of the Artificial Bee Colony (ABC) Algorithm, *Recent advances in neural networks, fuzzy systems & evolutionary computing*, WSEAS Press, June 2010, pp. 191-196

[14] Guerra L., Murino T., Romano E., A heuristic algorithm for the constrained location - routing problem, *International Journal of Systems Applications, Engineering & Development*, Vol. 1, No. 4, 2007, pp. 146-154

[15] S. Masrom, Siti Z. Z. Abidin, P. N. Hashimah, A. S. Abd. Rahman, Providing flexibility in the Vehicle Route Optimization for the AVL based Transport Monitoring System, *WSEAS Transactions on Computers*, Vol. 9, Issue 5 2010, pp. 526-535

[16] Bodin L, Golden B., Assad A., Ball M., Routing and scheduling of vehicles and crews: The state of the art, *Computers and Operations Research*, Vol. 10, No. 2, 1983, pp. 63-211

[17] Deb K., An Efficient Constraint-handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, Issue 2-4, 2000, pp. 311-338

[18] Eiben, E., Smith, J. E., *Introduction to evolutionary computing*, Natural computing series, Springer, 300 p., 2003

[19] Vehicle Routing Data Sets, 2003, URL http://www.branchandcut.org/VRP/data

ISBN: 978-960-474-297-4