International Journal of
Intelligent Systems and Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

A Bee Colony Optimization-based Approach for Binary Optimization

Mustafa Servet Kıran *[a], Mesut Gündüz [a]

Received 11th September 2013, Accepted 21th September 2013

Abstract: *The bee colony optimization (BCO) algorithm, one of the swarm intelligence algorithms, is a population based iterative search algorithm. Being inspired by collective bee intelligence, BCO has been proposed for solving discrete optimization problems such as travelling salesman problem. The BCO uses constructive approach for creating a feasible solution for the discrete optimization problems but in this study, we used the solution improvement technique due to nature of the uncapacitated facility location problem (UFLP). In the proposed method named as binBCO, the feasible solutions are generated for the artificial bees in hive of BCO and these solutions are tried to improve by utilizing interaction in the hive. At the end of the each iteration, some of the bees leave self-solutions and the leaving process depends on the loyalty of the bee to the self-solution. After a bee leaves self-solution, a random feasible solution is generated and assigned to this bee. In order to show the performance of binBCO, we examined it on well-known UFLPs, and the experimental studies show that the proposed method produces promising results.*

Keywords: *Bee colony optimization, Binary optimization, Uncapacitated facility location problem.*

1. Introduction

In recent years, many swarm intelligence-based optimization algorithms have been proposed in the literature, such as ant colony optimization [1,2], particle swarm optimization [3], bee colony optimization [4], artificial bee colony [5]. The BCO algorithm was proposed for solving travelling salesman problem which is a combinatorial optimization problem. In the basic BCO concepts, the partial tours are created, and while these partial tours are constructed, interactions such as information sharing, tour sharing are used. In this study, BCO, a constructive method for solving travelling salesman problem, is modified for binary optimization and transformed to path improvement method due to the fact that the objective function value of partial solutions cannot be calculated in dealing problem in this study.

Since invention of BCO in 2001, many discrete optimization problems have been resolved by this technique. The BCO was first applied to solve travelling salesman problem [6-8], and BCO is used for solving vehicle routing problem with uncertainty demand [9], ride matching problem [10,11], travelling salesman problem and a routing problem in networks [12], routing and wavelength assignment in all-optical networks [13], static scheduling of independent tasks on homogeneous multiprocessor systems [14], driver-line-time scheduling [15], p-center problem [16]. A detailed literature review about bee colony optimization and the other bee colony-based swarm intelligence techniques can be found in [4, 17].

Based on the literature review, BCO algorithm is applied to solve many discrete optimization problems. In this study, BCO algorithm is used for solving uncapacitated facility location problem by developing a binary version the BCO algorithm. The rest of paper is organized as follows. The mathematical background and objective of UFLP, BCO algorithm and proposed method are explained in the Section 2. The results of the simulations are given in Section 3 and the results are discussed in Section 4. Finally, conclusion and future works are presented in Section 5.

2. Material and Methods

2.1. Mathematical Model of Uncapacitated Facility Location Problem

A brief description about UFLP which is used to show the performance and accuracy of the proposed binBCO algorithm is given below. An UFLP with *i* candidate facility and *j* customers sites can be represented by a network with *i+j* nodes and *ij* arcs. In the UFLP, f_i is used to represent the cost of opening facility *i*, and c_{ij} is used to represent the cost of serving customer *j* from facility *i* or assigning customer *j* to facility *i*. It is assumed that $c_{ij} \geq 0$ for all *i*=1,2,…*I* and *j*=1,2,…*J* and $f_i \geq 0$ for all *i*=1,2,…*I* [18] .x_{ij} is the continuous variable representing the amount supplied to customer *j* from facility *i* and y_i is the 0-1 variable such that y_i= 1 if facility *i* is established and 0 otherwise [19]. The solution process of the UFLP is to find an optimal solution that satisfies all customer demand and minimizes the total cost (Eq. 23). The UFLP can be formally stated as [20]:

$$f(UFLP) = min\{\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i\} \quad (1)$$

Subject to:

$$\sum_{i\in I} x_{ij} = 1,\ j \in J \quad (2)$$

$$x_{ij} \leq y_i,\ i \in I \ and\ j \in J \quad (3)$$

[a] *Selcuk University, Faculty of Engineering, Department of Computer Engineering*
** Corresponding Author: Email: mskiran@selcuk.edu.tr;*

$$x_{ij} \in \{0,1\},\ i \in I \ and\ j \in J \quad (4)$$

$$y_i \in \{0,1\},\ i \in I \quad (5)$$

Constraint (2) makes sure that all demands have been met by a particular open site, and constraints (3)-(5) is to keep variables integer. With this model, we have to decide (*i*) the number of facility sites to be established and (*ii*) the quantities to be supplied from facility *i* to customer *j* such that the total cost (comprising fixed and variable costs) is minimized. When the locations of facilities which will be opened are determined, which customer will be served by a facility can be obtained easily. Therefore, we address a vector *A* of *n* variables where each variable in *A* is bit. Each bit (A_i, i=1,2,…,n) indicates that whether a facility is opened at location *i* or not. Due to the fact that the solution of each bee represents this vector, this vector plays a critical role in binBCO.

Location problems are some of the most widely studied problems in combinatorial optimization [21]. The UFLP which is one of the main problems in location problems is NP-hard even it has an uncapacitated assumption. This is perhaps the most common location problem, having been widely studied in the literature, both in theory and in practice [22]. Exact algorithms for UFLP do exist such as the dual approach of Erlenkotter [23], the primal-dual approaches of Körkel [24] and the branch-and-bound approach of Galvão and Raggi [25] but its NP-hard nature and high computation time makes heuristics and meta-heuristics the natural choice for larger instances. For this reason, some approximate methods have been proposed for solving UFLPs. These methods cannot guarantee to find the optimal solution but can obtain optimum or near optimum solution in a reasonable time. The meta-heuristics used to solve UFLPs are mainly genetic algorithm by Topçuoğlu et al. [26], tabu search by Sun [18], continuous and discrete particle swarm optimization algorithms by Sevkli and Güner [27] and Guner and Sevkli [28], simulated annealing by Yiğit et al. [20], artificial bee colony (ABC) algorithm by Kıran and Gündüz [29].

2.2. Bee Colony Optimization

The basic BCO was developed by inspiring natural behaviors of real bees between nest and food source. In the BCO, all bees try to construct a feasible solution for the optimization problems.

1. Determine the number of bees in the hive. (NB)
2. Assign the empty solutions to the artificial bees.
3. Determine the stopping condition for the algorithm.(SC)
4. While SC is met
   a. Forward Pass: Allow the bees to construct self-partial tours.
   b. Backward Pass: Return all bees to the hive and evaluate quality of partial solutions of bees. Test the loyalty of the bees to self-solutions and if any bee abandons self-solution, select a solution of recruiters (loyal bee to self -solutions) and assign this solution to the bee which abandons self-solution.
   c. If the partial solutions of the bees are completed, evaluate the all solutions and determine the best one and clear the memory of the bees (delete the all solutions)

Report the best solution found by the bee population.

Fig.1. The basic steps of the BCO algorithm

At beginning of the search process, all bees are in the hive and the BCO consists of two phases sequentially realized. First phase is forward pass and in this phase, the bees create self-partial solution for the problem. The second phase is backward pass and in this phase, the bees that they are in the search space return to the hive in order to share their information about the partial solutions found. Being dependent on loyalty of the bees to the self-solutions, some of the bees leave self-partial solutions, and the partial solutions of the other bees are assigned for these bees. This process is named as recruitment process. After the partial tours are completed, an iteration of BCO is finished. The best solution in the population is determined and saved, and all solutions in the memory of bees are deleted in the basic version of BCO algorithm [4]. After the explanations are given above, the basic steps of the algorithm are presented in Fig. 1.

2.3. Proposed Method

In order to solve the UFLP, BCO algorithm is transformed to path improvement technique in the proposed method. The main difference between basic BCO and proposed approach is that the feasible solutions are assigned to the bees at the beginning of the algorithm. In the binBCO, the bees try to improve the self-solutions at the each iteration in the forward and backward passes instead of solution construction. After the bee population is created with feasible solutions for the UFLP, the best solution in the population is found. For each solution, objective function value (by Eq.1) and fitness of the solution by using Eq. 6 given below are calculated [30].

$$fit_i = \frac{1}{1+obj_i} \quad , i = 1,2,...,NB \quad (6)$$

where, fit_i is the fitness of the i^{th} solution (or bee), obj_i is the objective function value of i^{th} solution and *NB* is the number of bees.

In the forward pass of the binBCO, the new solutions are created for the bees to improve the solutions of the bees given as follows:

$$P = \frac{bestFit}{fit_i + bestFit},\ i = 1,2,...,NB \quad (7)$$

$$Pb_i = \frac{fit_i}{fit_i + bestFit} \quad (8)$$

$$B_{i,j} = \begin{cases} Best_j, & if\,(rnd_{i,j} < P) \\ B_{i,j}, & otherwise \end{cases} \quad (9)$$

where, *P* is the selection probability of the best solution, Pb_i is the selection probability of the bee, $bestFit$ is the fitness of the best solution in the population, $B_{i,j}$ is the j^{th} decision variable of the i^{th} solution, $Best_j$ is the j^{th} decision variable of best solution in the population and $rnd_{i,j}$ is a random number in range of [0,1]. In order to show improvement of the solution, an illustrative example is given below.

Assume that the best solution in the population is 011010, i^{th} solution (B_i) is 111001, the fitness of the best is 0.7 and the fitness of the B_i is 0.5. According to Eq.7, the selection probability of the best solution is calculated as 0.58, and the selection probability of the B_i solution is calculated 0.42 according to Eq.8. Assume that the random numbers generated for each dimension are 0.2, 0.9, 0.4, 0.3, 0.7, 0.6. According to these assumptions, the new solution is obtained as follows:

B_i

1	1	1	0	0	1

Best

0	1	1	0	1	0

Random Numbers

0.2	0.9	0.4	0.3	0.7	0.1

New Solution for B_i

0	1	1	0	0	0

After new solutions are obtained for each bee, the loyalty of the bees to the solutions is calculated as follows:

$$L_i = \frac{fit_i}{\sum_{i=1}^{NB} fit_i} \tag{10}$$

where, L_i is the loyalty of i^{th} bee to self-solution and this is used for determining whether a bee abandon self-solution or not. If the loyalty of the bee is less than the mean loyalty of the population (obtained using Eq.11, where L_p is the mean loyalty value of the population), the bee abandons self-solution, and a random solution is generated for this bee.

$$L_p = \frac{1}{NB} \times \sum_{i=1}^{NB} fit_i \tag{11}$$

where, L_p is the loyalty of the population and it is used to test the loyalty of the bees to self-solution. After proposed method is explained above, the algorithmic framework of the binBCO is presented in Fig.2.

1. Determine the number of bees in the hive. (NB)
2. Generate feasible solutions (binary values) for the bees.
3. Calculate objective function value for each bee.
4. Calculate fitness of the solutions of the bees by using Eq.6.
5. Determine the best solution in the population and save it as global best.
6. While a stopping condition is met
   a. For each bee
      i. Calculate selection probabilities of the best solution and the solution of bee by using Eq. 7 and 8.
      ii. For each decision variable
         1. If generated random number is less than the selection probability of best solution, the decision variable comes from the best solution, otherwise self-solution.
   b. Calculate objective function values and fitness of the solutions.
   c. Calculate the loyalty bees and population by using Eq.10 and Eq. 11.
   d. Fix the bees that they are not loyal to self-solution.
   e. Generate random solution for these bees.
   f. Determine the best solution in the population.
   g. If the best solution in population is better than the global best, replace them.
7. Report the best solution found by the population.

Fig.2. The binBCO algorithm

3. Computational Experiments

We examined the performance and accuracy of binBCO algorithm on the small and medium size test problems taken from OR-Library. The description for the problems is given in Table 1. Cap71-74 are the small size problems and they contain 16 decision variables. The rest of test suite is medium size problems and they contain 50 decision variables, and also the costs of the optimal solutions are presented in Table 1.

There are three control parameters in the binBCO algorithm. The first control parameter is the population size and it is analyzed by using different values such as 20, 40, 60, 80, 100. The second control parameter is the diversification ratio (DR). This parameter controls how many bees abandons self-solution at the each iteration. As previously mentioned, if the loyalty of a bee is less than the mean loyalty of population, this bee abandons self-solution, and a random solution is generated for this bee. In order to prevent much diversification in the population, DR parameter is set to 0.4 and it means that the random solutions are generated for 40% of bees that they abandoned self-solutions. If obtained number is floating number, it is rounded the nearest integer number. The last control parameter is the stopping condition for the binBCO algorithm. We used maximum iteration number for the stopping condition and it is set to 1000.

Table 1. The description of the test suite

Problem Name	Problem Size	Cost of Optimal Solution
Cap71	16 × 50	932615.75
Cap72	16 × 50	977799.40
Cap73	16 × 50	1010641.45
Cap74	16 × 50	1034976.98
Cap101	25 × 50	796648.44
Cap102	25 × 50	854704.20
Cap103	25 × 50	893782.11
Cap104	25 × 50	928941.75
Cap131	50 × 50	793439.56
Cap132	50 × 50	851495.33
Cap133	50 × 50	893076.71
Cap134	50 × 50	928941.75

Under these conditions, the binBCO algorithm is coded on Matlab platform, and the experiments are run on a PC with Intel i5 3.1 Ghz microprocessor and 4 Gb Ram. The proposed method is run 10 times for each problem and obtained results are presented in Table 2, 3 and 4. In the results tables (Table 1, 2 and 3), the best solution for each problem obtained by binBCO is written as bold font type. As seen from the result tables, while the population size is 100, better results is generally produced by the binBCO.

4. Results and Discussion

In this study, we proposed a binary version of the basic BCO algorithm, called as binBCO, and promising results for UFLPs were produced by the binBCO algorithm. While the number of decision variables or problem dimensionality is increased, the performance of the binBCO algorithm is decreased. In order to improve the local search ability of the binBCO algorithm, new local search techniques should be used. The other issue is to control diversification in the population. The random solutions are generated for the bees that they are not loyal self-solutions, this has caused more diversification in the population, and the intensification of the population has been weakened. In order to balance diversification in the population, we proposed DR control parameters for the bees that they are not loyal to self-solutions.

5. Conclusion and Future Works

The basic BCO algorithm was inspired by the intelligent behavior of real bees between nest and food sources. Increasing the popularity of BCO in solving the optimization problems shows that BCO algorithm is competitive method. In this study, a binary version of the basic BCO algorithm is developed and the solution construction approach is changed as solution improvement. The performance and accuracy of the new proposed method are tested on the UFLP which is a pure binary optimization problem. Experimental simulations show that proposed approach produces promising results. In our future works, we will include local search methods to the binBCO and compare the performance of binBCO with the other binary optimization methods.

Table 2. The results obtained under population sizes 20 and 40

Problem	Pop_Size=20				Pop_Size=40			
	Worst	Mean	Best	Std.Dev.	Worst	Mean	Best	Std.Dev.
Cap71	**932615.8**	**932615.8**	**932615.8**	**0**	935152.3	932869.4	932615.8	802.1
Cap72	981649.4	978292.1	977799.4	1227.2	982711.6	978290.6	977799.4	1553.4
Cap73	1014491.4	1011760.7	1010641.5	1323.8	**1012477**	**1011192.1**	**1010641.5**	**886.6**
Cap74	1037717.1	1035251	1034977	866.5	**1034977**	**1034977**	**1034977**	**0**
Cap101	801588.4	798688.6	796648.4	1538.9	800769.6	798229.1	796648.4	1241.7
Cap102	857956.1	856846	856004.4	669.6	859963.3	855686.6	854704.2	1612.7
Cap103	901977.1	895224	893782.1	2433.5	900022.1	895552.3	893782.1	2289
Cap104	935591.7	932481.3	928941.8	3062.6	942072.6	932111.5	928941.8	4563.7
Cap131	816940.8	807571.7	802160.2	4488.6	818222.8	806566.5	797306.9	6438.4
Cap132	887085.1	870204.9	861570.6	8548.9	878232.8	868713.4	854879	8269.9
Cap133	950367.4	918466.1	895642.5	16230.7	926371	910563.4	899963.4	8979
Cap134	983826.1	963197.5	945187.6	12522.8	965165.3	940782.6	934573.8	9367.2

Table 3. The results obtained under population sizes 60 and 80

Problem	Pop_Size=60				Pop_Size=80			
	Worst	Mean	Best	Std.Dev.	Worst	Mean	Best	Std.Dev.
Cap71	933568.9	932806.4	932615.8	401.9	**932615.8**	**932615.8**	**932615.8**	**0**
Cap72	981649.4	978569.4	977799.4	1623.3	982711.6	978290.6	977799.4	1553.4
Cap73	1014253.4	1011553.3	1010641.5	1284.1	1012477	1011559.2	1010641.5	967.4
Cap74	1037717.1	1036621	1034977	1415	1037717.1	1035525	1034977	1155.3
Cap101	800005	798245	796648.4	1201.9	800005	797826.2	796648.4	1176.9
Cap102	**858250.6**	**855629.6**	**854704.2**	**1334.2**	861025.5	856448.4	854704.2	1923.7
Cap103	898551.5	895440.6	894008.1	1897.9	898551.5	894915.5	893782.1	1447.9
Cap104	934587	930941.9	928941.8	2687.4	**934587**	**930635.3**	**928941.8**	**2726.9**
Cap131	809812	803524.7	797205.9	4721.1	812891.8	805886.4	799218.7	4934.4
Cap132	871487.2	863067.2	857054	4509.6	867959.6	858896.3	851495.3	5576.7
Cap133	909133.4	900254.7	894752	5110.4	**905245.8**	**899765.4**	**893733**	**4012.8**
Cap134	985002	956864.6	929477.6	21159.7	956707	938263.8	928941.8	8831.5

Table 4. The results obtained under population size 100

Problem	Pop_Size=100			
	Worst	Mean	Best	Std.Dev.
Cap71	**932615.8**	**932615.8**	**932615.8**	**0**
Cap72	**981649.4**	**978184.4**	**977799.4**	**1217.5**
Cap73	1012477	1011375.7	1010641.5	947.9
Cap74	**1034977**	**1034977**	**1034977**	**0**
Cap101	**799144.7**	**797803.2**	**796648.4**	**1168.6**
Cap102	861111.5	856276.3	854704.2	1913.5
Cap103	**898551.5**	**894859**	**893782.1**	**1411.4**
Cap104	934587	931764.4	928941.8	2975.3
Cap131	**804460.4**	**800422.8**	**794373.4**	**3250.5**
Cap132	**865076.8**	**858437.4**	**852762.9**	**3353.9**
Cap133	906857.7	900166.6	894095.8	3768.1
Cap134	**950386.9**	**937825.8**	**929477.6**	**6341**

References

[1] Dorigo M., Maniezzo V., Colorni A. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics -Part B 26: 29–41, 1996.

[2] Dorigo M., Gambardella L. M., Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1:53–66, 1997.

[3] Eberhart RC., Kennedy J, A new optimizer using particle swarm theory. In Proc. of Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43, 1995.

[4] Teodorovic D., Bee colony optimization (BCO). Eds: Lim C.P., Jain L.C., Dehuri S., Innovations in swarm intelligence, Berlin, Heidelberg, Springer-Verlag, 39-60, 2009.

[5] Karaboga D,. An idea based on honey bee swarm for numerical optimization, Erciyes University, Technical Report-TR06, Kayseri/Turkey, 2005.

[6] Lucic P., Teodorovic D,. 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, 441-445, 2001.

[7] Lucic P., Teodorovic D., Transportation Modeling: an artificial life approach. In Proc. of the 14th IEEE International Conference on Tools with Artificial Intelligence, Washington, 216-223, 2002.

[8] Lucic P., Teodorovic D., Computing with bees: attacking complex transportation engineering problems. International Journal of Artificial Intelligence Tools 12: 375-394, 2003.

[9] Lucic P., Teodorovic D., Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. Ed.: Verdegay J.L., Fuzzy Sets in Optimization, Springer-Verlag, Heidelberg, Berlin, 67-82, 2003.

[10] Teodorovic D., Dell'Orco M., Bee colony optimization-a cooperative learning approach to complex transportation problems. In Proc. of the 10th Meeting Euro Working Group on Transportation, Poznan, Poland, 51-60, 2005.

[11] Teodorovic D., Dell'Orco M., Mitigating traffic congestion: solving the ride matching problem by bee colony optimization. Transportation Planning and Technology 31: 135-152, 2008.

[12] Teodorovic D., Lucic P., Markovic G., Dell'Orco M., Bee

colony optimization: Principles and Applications. Eds.: Reljin B., Stankovic S., In Proc. of the 8th Seminar on Neural Network Applications in Electrical Engineering – NEUREL 2006, Belgrade, 151-156, 2006.

[13] Markovic GZ., Teodorovic DB., Raspopovic VSA., Routing and wavelength assignment in all-optical networks based on the bee colony optimization. AI Communications 20: 273-285, 2007.

[14] Davidovic T., Selmic M., Teodorovic D., Scheduling Independent Tasks: Bee Colony Optimization Approach. In Proc. of the 17th Mediterranean Conference on Control&Automation, Thessaloniki, Greece, 1020-1025, 2009.

[15] Kıran MS., Bee Colony-based Driver-Line-Time Optimization. Master Thesis (In Turkish), 2010.

[16] Davidovic T., Ramljak D., Selmic M., Teodorovic D., Bee colony optimization for the p-center problem. Computers and Operations Research 38:1367-1376, 2011.

[17] Tedorovic D. Swarm intelligence systems for transportation engineering: Principles and applications. Transportation Research Part C: Emerging Technogies 16: 651-667, 2008.

[18] Sun M. Solving the uncapacitated facility location problem using tabu search. Computers & Operations Research 33: 2563-2589, 2006.

[19] Tcha D-W., Myung Y-S., Chung K-H. Parametric uncapacitated facility location. European Journal of Operational Research 86: 469-479, 1995.

[20] Yiğit V., Aydın ME., Türkbey O., Solving large-scale uncapacitated facility location problems with evolutionary simulated annealing. International Journal of Production Research 44: 4773-4791, 2006.

[21] Ghosh D., Neighborhood search heuristics for the uncapacitated facility location problem. European Journal of Operational Research 150: 150-162, 2003.

[22] Resende MGC., Werneck R.F., A hybrid multistart heuristic for the uncapacitated facility location problem. European Journal of Operational Research 174: 54-68, 2003.

[23] Erlenkotter D., A dual-based procedure for uncapacitated facility location. Operations Research 26: 992–1009, 1978.

[24] Körkel M., On the exact solution of large-scale simple plant location problems. European Journal of Operational Research 39: 157–73, 1989.

[25] Galvao RD., Raggi LA. A method for solving to optimality uncapacitated location problems. Annals of Operations Research 18: 225-244, 1989.

[26] Topçuoğlu H., Corut F., Ermiş M., Yılmaz G., Solving the uncapacitated hub location problem using genetic algorithms. Computers & Operations Research 32: 967-984, 2005.

[27] Sevkli M., Guner AR., A continuous particle swarm optimization algorithm for uncapacitated facility location problem. In Proc. of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence - ANTS 2006, Brussels, Belgium, 316-323, 2006.

[28] Guner AR., Sevkli M, A discrete particle swarm optimization algorithm for uncapacitated facility location problem. Journal of Artificial Evolution and Applications pp. 1-9, 2008.

[29] Kıran MS., Gündüz M.. XOR-based artificial bee colony algorithm for binary optimization, Turkish Journal of Electrical Engineering and Computer Sciences, doi: 10.3906/elk-1203-104, 2013.

[30] Karaboga D., An idea based on honey bee swarm for numerical optimization, Erciyes University, Technical Report-TR06, Kayseri/Turkey, 2005.