A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem

Dervis Karaboga
Department of Computer Engineering
Erciyes University
Kayseri, TURKEY
karaboga@erciyes.edu.tr

Beyza Gorkemli
Department of Computer Engineering
Erciyes University
Kayseri, TURKEY
bgorkemli@erciyes.edu.tr

Abstract— **Traveling Salesman Problem is an important optimization issue of many fields such as transportation, logistics and semiconductor industries and it is about finding a Hamiltonian path with minimum cost. To solve this problem, many researchers have proposed different approaches including metaheuristic methods. Artificial Bee Colony algorithm is a well known swarm based optimization technique. In this paper we propose a new Artificial Bee Colony algorithm called Combinatorial ABC for Traveling Salesman Problem. Simulation results show that this Artificial Bee Colony algorithm can be used for combinatorial optimization problems.**

Keywords-component; artificial bee colony; combinatorial optimization; traveling salesman problem

I. INTRODUCTION

Traveling Salesman Problem (TSP) is an NP-hard combinatorial optimization problem [1], so in the literature a lot of metaheuristic algorithms have been applied to this problem for obtaining better results in acceptable computational times. Some of these algorithms include ant colony optimization (ACO) and its variants: Dorigo and Gambadella applied the ant colony system to TSP in [2], Zhang and Feng used a Max-Min ant system for TSP in [3], Ilie and Badica proposed a distributed approach for ACO algorithms on a distributed architecture in [4], Gan, Guo, Chang and Yi also proposed an ACO algorithm for TSP in [5] and they used a scout characteristic to solve the stagnation behavior and premature convergence problem of the basic ACO algorithm on TSP, Puris, Bello and Herrera used a Two-Stage approach to improve the quality of the solutions of ACO [6]. Wong, Low and Chong described a Bee Colony Optimization (BCO) algorithm for TSP in [7]. Then they integrated this algorithm with 2-opt heuristic to further improve prior solutions generated by the BCO model [8]. Kim and Cho applied a hybrid cultural algorithm with local search to TSP in [9]. Many other researchers developed bio-inspired methods including Simulated Annealing [10], Tabu Search [11, 12], Particle Swarm Optimization [13, 14], Self Organizing Map [15], Genetic Algorithms [16, 17] for TSP. In addition to these methods, to find better results combinations of these methods especially hybridization with local search heuristic algorithms have been widely used [17, 18, 19].

Artificial Bee Colony (ABC) algorithm is a nature inspired algorithm that was introduced by Karaboga for multi-modal and multi-dimensional numeric problems [20, 21]. ABC algorithm gives good results for continuous optimization problems [22, 23]. A few discrete versions of ABC algorithm have been introduced into literature [24, 25]. In this paper we propose a new combinatorial ABC algorithm to solve TSP. We organized the rest of the paper as follows: Section 2 explains TSP, the proposed approach is presented in section 3 and followed by the simulation results and the comparison of Genetic Algorithm and combinatorial ABC algorithm for this problem in section 4. Finally, section 5 concludes the paper and discusses the future path of our work.

II. TRAVELING SALESMAN PROBLEM

The basic principle of TSP is that, the salesman starts from a point to his tour and he returns to this starting point as trying to obtain a closed tour with minimum cost. When he takes his tour he must visit every point once. The cost of his tour directly depends on the tour length.

In this study, the distance between the city i and city $(i+1)$ $d(T[i], T[i+1])$ is calculated as Euclidean distance by (1).

$$d(T[i], T[i+1]) = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (1)$$

Total tour length f can be expressed as fallows (2)

$$f = \sum_{i=1}^{n-1} d(T[i], T[i+1]) + d(T[n], T[1]) \qquad (2)$$

where n is the total number of cities.

III. COMBINATORIAL ARTIFICIAL BEE COLONY ALGORITHM

ABC algorithm is a swarm based metaheuristic that simulates foraging behavior of honey bees. The artificial bee colony includes three kinds of bees considering the division of labor: employed bees, onlooker bees and scout bees. Each employed bee works on only one food source. An employed bee keeps a food source in her mind when she leaves from the hive and she shares the information about her food source with onlookers on dance area. Onlookers select a food source by watching the dances of the employed bees and try to improve this source. If a food source is abandoned, its employed bee becomes a scout to explore new food sources randomly. Basic steps of the ABC algorithm are given below:

Initialize

REPEAT

- Employed bees phase: send the employed bees to their food sources and determine their nectar amounts.
- Onlooker bees phase: send onlookers to the food sources depending on their nectar amounts and determine their nectar amounts.
- Scout bees phase: send scouts to search new food sources that are replaced with abandoned ones.
- Memorize the best food source found so far.

UNTIL (termination criteria is satisfied)

In ABC algorithm each food source represents a possible solution for the problem and the fitness value of the solution corresponds to the nectar amount of this food source.

Our new combinatorial version of the ABC algorithm, CABC, employs the same basic steps presented above. The aim of TSP optimization is minimizing the total closed tour length. So the quality of each solution fit_i is calculated by using (3).

$$fit_i = \frac{1}{1 + f(x_i)} \tag{3}$$

where $f(x_i)$ is the tour length of the solution x_i.

For the scout limit value, l is calculated by (4).

$$l = \frac{cs \times D}{3} \tag{4}$$

where cs represents the colony size and D stands for the dimension of the problem ($D = n$).

Detailed steps of CABC algorithm are fallows:

1. Initialize the parameters: colony size cs, maximum number of iterations *MaxNumber*.
2. Initialize the positions of the food sources x_i. $i = 1, 2, \ldots, cs$.
3. Evaluate the population of solutions.
4. Memorize the best solution.
5. $c = 0$
6. REPEAT
   - *Employed bees phase:* for each employed bee;
     - Produce a new solution υ_i in the neighborhood of x_i and evaluate it.
     - Apply the greedy selection process between x_i and υ_i.
   - Calculate the probability values P_i for the solutions x_i by means of their fitness values by using (5).

$$P_i = \frac{0.9 \times fit_i}{fit_{best}} + 0.1 \tag{5}$$

   - *Onlooker bees phase:* for each onlooker;
     - Produce a new solution υ_i from the solution x_i, selected depending on P_i and evaluate it.
     - Apply the greedy selection process between x_i and υ_i.
   - Memorize the best solution achieved yet.
   - Determine the abandoned solution, if exists, replace it with a new produced solution for the scout.
   - $c = c + 1$
7. UNTIL c = *MaxNumber*.

Recently, Albayrak and Allahverdi developed a new mutation operator to increase Genetic Algorithm's performance to find the shortest distance in TSP [26]. They tested GSTM operator on some benchmark TSPs and compared the results with other seven mutation operators. Regarding to the best and average error values, GSTM produced the best results among these operators. So in our study, the GSTM operator was adapted to the neighborhood search mechanism of employed and onlooker bees. Steps of this mechanism are:

1. Select solution x_k randomly in the population. ($k \neq i$)
2. Select a city $x_i[j]$ randomly.
3. Select the value of searching way parameter Φ randomly. ($\Phi = \{-1, 1\}$)
4. If ($\Phi = -1$) then

   The city visited before the $x_k[j]$ is set as the before city of $x_i[j]$.

   Else

   The city visited after the $x_k[j]$ is set as the after city of $x_i[j]$.

   End if // This operation obtains a new closed tour $T^{\#}$.
5. As a result of this new connection there will be an open sub tour T^*. This sub tour's first city is assigned as R_1, last city is assigned as R_2.
6. If ($random \leq P_{RC}$) then

   Add T^* to $T^{\#}$ so that there will be minimum extension.

   Else

   If (random $\leq P_{CP}$)

   - Add each city of T^* to the T starting from the position R_1 by rolling or mixing with P_L probability.

   Else

   - Select randomly one neighbor from neighbor lists for the points R_1 and R_2 (NL_{R1} and NL_{R2})

- Invert the points NL_{R1} or NL_{R2} that provides maximum gain such a way that these points will be neighbors to the points R_1 or R_2.
- Repeat if the inversion is not taken place.

End if

End if

where $x_i[j]$ notation represents the position of the city j in solution i. *random* is a random generate number in (0,1). T is the original tour that represents the solution x_i without any neighborhood searches. GSTM parameters: P_{RC} is reconnection probability, P_{CP} is correction and perturbation probability, P_L is linearity probability and minimum sub tour length is given as L_{MIN}, maximum sub tour length is L_{MAX}, neighborhood list size is NL_{MAX}.

Gain of the point R G_R is calculated with (6).

$$\begin{aligned} G_R = & \; d[T[R],\, T[R+1]] + d[T[NL_R],\, T[NL_R+1]] \\ & - (d[T[R],\, T[NL_R]] + d[T[R+1],\, T[NL_R+1]]) \end{aligned} \tag{6}$$

IV. SIMULATION RESULTS

Performance of CABC was tested on 2 benchmark problems: one of them is KroB150 that contains 150 cities and the other is KroA200 with 200 cities. These problems data can be found in the TSPLIB: http://www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

10 runs were carried out for each problem with random seeds. For comparing the results fairly, we took the same parameter values used in [26] as shown in Table 1. The simulation results and optimum tour lengths for these two problems are shown in Table 2.

TABLE I. PARAMETER SETTINGS

Parameter	Value
cs	40
MaxNumber	20000
P_{RC}	0.5
P_{CP}	0.8
P_L	0.2
L_{MIN}	2
L_{MAX}	Int(sqrt(n))
NL_{MAX}	5

TABLE II. OPTIMAL TOUR LENGTH AND THE BEST RUN FOUND BY CABC

Problem	Optimal Tour Length	CABC (best)
KroB150	26130	26247
KroA200	29368	29420

Initial solutions were produced by Nearest Neighbor tour construction heuristic as in [26] and distances between the cities were measured by integer numbers. Table 3 shows a comparison of CABC algorithm and genetic algorithm employing the eight different genetic mutation operators [26].

TABLE III. PERFORMANCE COMPARISON OF GA AND CABC ALGORITHM

Methods	KroB150		KroA200	
	Best Error	*Ave. Error*	*Best Error*	*Ave. Error*
EXC	2.9277	5.0919	2.5061	4.9877
DISP	7.8263	10.4635	6.7284	8.8276
INV	7.1183	9.0521	6.2245	8.3594
INS	1.8178	4.5488	2.1554	4.2819
SIM	2.8397	3.8596	1.5766	3.2457
SCM	3.2836	7.0609	3.9873	5.4502
GSM	3.4520	4.9541	4.4675	5.7018
GSTM	0.9644	1.7616	0.8683	1.5432
CABC	0.4478	0.9847	0.1771	0.6211

V. CONCLUSIONS

In this paper, a new combinatorial version of ABC algorithm is introduced for solving TSP. Experimental results show that this approach gives good solutions for the considered two examples of this NP-hard combinatorial optimization problem. As a future work, the algorithm will be applied on more complex test problems and this algorithm will be hybridized with some local search heuristics to find better results.

REFERENCES

[1] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," European Journal of Operational Research, vol. 59, no. 2, pp. 231-247, 1992.

[2] M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approachto the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 53-66, April 1997.

[3] Z. Zhang, and Z. Feng, "A novel max-min ant system algorithm for traveling salesman problem," in Proceedings of Intelligent Computing and Intelligent Systems (ICIS 2009), 2009, pp. 508-511.

[4] S. Ilie, and C. Badica, "Effectiveness of solving traveling salesman problem using ant colony optimization on distributed multi-agent middleware," in Proceedings of the International Multiconference on Computer Science and Information Technology, 2010, pp.197-203.

[5] R. Gan, Q. Guo, H. Chang, and Y. Yi, "Improved ant colony optimization algorithm for the traveling salesman proplems," Journal of Systems Engineering and Electronics, vol. 21, no. 2, pp. 329-333, April 2010.

[6] A. Puris, R. Bello, and F. Herrera, "Analysis of the efficacy of a two-stage methodology for ant colony optimization:case of study with TSP and QAP," Expert Systems with Applications, vol. 37, no. 7, pp. 5443-5453, July 2010.

[7] L. P. Wong, M. Y. H. Low, and C. S. Chong, "A bee colony optimization algorithm for traveling salesman problem," in Proceedings of Second Asia International Conference on Modelling & Simulation (AMS 2008), 2008, pp. 818-823.

[8] L. P. Wong, M. Y. H. Low, and C. S. Chong, "Bee colony optimization with local search for traveling salesman problem," in Proceedings of Industrial Informatics (INDIN 2008), 2008, pp. 1019-1025.

[9] Y. Kim, and S. B. Cho, "A hybrid cultural algorithm with local search for traveling salesman problem," in Proceedings of Computational Intelligence in Robotics and Automation (CIRA 2009), 2009, pp. 188-192.

[10] J.W. Pepper, B.L. Golden and E.A. Wasil, "Solving the traveling salesman problem with annealing-based heuristics: a computational study," IEEE Transactions on Systems, Man and Cybernetics-Part A, vol.32, pp.72-77, 2002.

[11] Y. He, Y. Qiu, G. Liu and K. Lei, "A parallel adaptive tabu search approach for traveling salesman problems," in Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005, pp.796-801.

[12] N. Yang, P. Li and B. Mei, "An angle-based crossover tabu search for the traveling salesman problem," in Proceedings of International Conference on Natural Computation, 2007, pp.512-516.

[13] W.L. Zhong, J. Zhang and W.N. Chen, "A novel discrete particle swarm optimization to solve traveling salesman problem," in Proceedings of IEEE Congress on Evolutionary Computation, 2007, pp.3283-3287.

[14] Z. Yuan, L. Yang, Y. Wu, L. Liao and G. Li, "Chaotic particle swarm optimization algorithm for traveling salesman problem," in Proceedings of IEEE International Conference on Automation and Logistics, 2007, pp.1121-1124.

[15] A. Zhu and S.X. Yang, "An improved self-organizing map approach to traveling salesman problem," in Proceedings of IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003, pp.674-679.

[16] J.D. Wei and D.T. Lee, "A new approach to the traveling salesman problem using genetic algorithms with priority encoding," in Proceedings of Congress on Evolutionary Computation, 2004, pp.1457-1464.

[17] W. Xuan and Y. Li, "Solving traveling salesman problem by using a local evolutionary algorithm," in Proceedings of the IEEE International Conference on Granular Computing, 2005, pp.318-321.

[18] C.M. White and G.G. Yen, "A hybrid evolutionary algorithm for traveling salesman problem," in Proceedings of the IEEE Congress on Evolutionary Computation, 2004, pp.1473-1478.

[19] B. Freisleben and P. Merz, "A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems," in Proceedings of International Conference on Evolutionary Computation, 1996, pp. 616-621.

[20] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[21] D. Karaboga, "Artificial bee colony algorithm," www.scholarpedia.org/article/Artificial_bee_colony_algorithm, Scholarpedia. 5(3):6915, 2010.

[22] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, vol. 8, no. 1, pp. 687-697, January 2008.

[23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) Algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, November 2007.

[24] Q. K. Pan, M. F. Tasgetiren, P. N. Suganthan and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," Information Sciences, in press.

[25] A. Singh, "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem," Applied Soft Computing, vol. 9, no. 2, pp. 625-631, March 2009.

[26] M. Albayrak, and N. Allahverdi, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," Expert Systems with Applications, vol. 38, no. 3, pp. 1313-1320, March 2011.