Quick Combinatorial Artificial Bee Colony -qCABC- Optimization Algorithm for TSP

Beyza Gorkemli

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

Dervis Karaboga

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

***Abstract*—Combinatorial Artificial Bee Colony Algorithm (CABC) is a new version of Artificial Bee Colony (ABC) to solve combinatorial type optimization problems and quick Artificial Bee Colony (qABC) algorithm is an improved version of ABC in which the onlooker bees behavior is modeled in more detailed way. Studies showed that qABC algorithm improves the convergence performance of standard ABC on numerical optimization. In this paper, to see the performance of this new modeling way of onlookers' behavior on combinatorial optimization, we apply the qABC idea to CABC and name this new algorithm as quick CABC (qCABC). qCABC is tested on Traveling Salesman Problem and simulation results show that qCABC algorithm improves the convergence and final performance of CABC.**

Keywords—combinatorial optimization; swarm intelligence; artificial bee colony

I. INTRODUCTION

Salesman Problem (TSP) is a basic combinatorial optimization problem of many fields such as logistics, transportation and semiconductor industries. This basic problem is about finding a Hamiltonian path with minimum cost and has an NP hard character [1]. Because of the high importance of getting better solutions in acceptable computational times, a lot of scientists applied metaheuristic algorithms to this problem. Some of these algorithms are ant colony optimization (ACO) based methods. In [2], Dorigo and Gambadella used the ant colony system to solve TSP. Zhang and Feng applied a Max-Min ant system to TSP in [3]. Gan et al. proposed an ACO algorithm for TSP in [4] and in this work a scout characteristic is used to solve the stagnation behavior and premature convergence problem of the basic ACO algorithm on TSP. Ilie and Badica improved a new distributed approach for ACO algorithms on a distributed architecture in [5]. Puris, Bello and Herrera applied a Two-Stage approach to improve the quality of the solutions of ACO [6]. Tuba and Jovanovic introduced an improved ant colony optimization algorithm with novel pheromone correction strategy [7]. Bee Colony Optimization (BCO) algorithm is also used for solving TSP by Wong, Low and Chong [8]. Then it is integrated with 2-opt heuristic to improve prior solutions generated by the BCO model [9]. Kim and Cho used a hybrid cultural algorithm with local search for TSP in [10]. Many other bio-inspired methods were developed including Tabu Search [11, 12], Simulated Annealing [13], Particle Swarm Optimization [14, 15], Genetic Algorithms [16, 17], Self Organizing Map [18] to solve TSP. For getting better results hybridizations of these methods especially with local search heuristic algorithms have been also widely used [19, 20, 21].

Artificial Bee Colony (ABC) algorithm is a swarm intelligence based nature inspired algorithm that was introduced by Karaboga for multi-modal and multi-dimensional numeric problems [22, 23]. In [24], a good survey study about ABC algorithm can be examined. Some of the researchers proposed the discrete versions of ABC for TSP. Li et al. described a discrete ABC algorithm that uses the concept of Swap Operator to solve TSP [25]. Li et al. developed an ABC algorithm for TSP by combining the modified nearest neighbor approach and the improved inver-over operation [26]. Combinatorial ABC (CABC) is one of the successful discrete versions of ABC algorithm and applied to TSP problems [27].

The authors introduced a new definition for the foraging behavior of onlooker bees in ABC [28]. In that study they modeled the onlooker bees' behavior more accurately than in the basic ABC and the simulation results showed that this new definition significantly improves the convergence performance of ABC. So, this new version of ABC was named as quick ABC (qABC). In this paper, we incorporate the new definition used in qABC to CABC algorithm used for the combinatorial problems. The rest of the paper is organized as follows; section 2 presents TSP and CABC algorithm is explained in section 3. Section 4 describes the proposed method and the simulation results are demonstrated in section 5 and in section 6, the conclusion is given.

II. TRAVELING SALESMAN PROBLEM

In a basic TSP, a salesman obtains a closed tour so that, he must visit every city once and his start and finish point are the same. While he obtains his closed tour, he tries to achieve minimum cost and the cost of his tour directly depends on the tour length.

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

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

This work was supported by Research Fund of the Erciyes University. Project Number: FBA-11-3536

Using the Euclidean distances between the cities, total tour length f can be expressed as fallows (2)

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

where n is the total number of cities.

III. COMBINATORIAL ARTIFICIAL BEE COLONY-CABC-ALGORITHM

CABC is a an algorithm introduced as a combinatorial variant of ABC optimization. In CABC, the basic steps of ABC algorithm were used and also, for an effective neighborhood solution searching mechanism, a successful mutation operator of genetic algorithm was adapted to ABC's neighborhood searching approach [27]. This mutation operator was introduced to the literature by Albayrak and Allahverdi [29].

In CABC algorithm, the steps of the employed and onlooker bees' neighbor production mechanisms 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 produces a new closed tour $T^{\#}$.
5. When this new connection is obtained there will be an open sub tour T^*. This sub tour's first city is assigned as R_1 and last city as R_2.
6. If ($random \leq P_{RC}$) then

    Add T^* to $T^{\#}$ so that a minimum extension is generated.

   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

$x_i[j]$ notation represents the position of the city j in solution i. *random* is a random generated 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 the 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}.

The point R's gain G_R is calculated by using (3).

$$\begin{aligned} G_R &= d[T[R], T[R+1]] + d[T[NL_R], T[NL_R+1]] \\ &- (d[T[R], T[NL_R]] + d[T[R+1], T[NL_R+1]]) \end{aligned} \tag{3}$$

IV. QUICK COMBINATORIAL ARTIFICIAL BEE COLONY -qCABC- ALGORITHM

In real honey bee colonies, employed bees and onlooker bees determine their food sources in different ways. Employed bees exploit the food sources which they have already visited before. However, an onlooker is affected by the dances of employed bees and decides a region of food sources to exploit, which she has not visited before. When she arrives at that region, she visually examines the sources from above and chooses the fittest one. In other words, she evaluates the information of sources which have a similarity in terms of position, before making a decision since she visits the region first time. However, in basic ABC algorithm, employed and onlooker bees determine their food sources by using the same formula. Therefore, Karaboga and Gorkemli introduced a new definition for the onlooker bees and described qABC in [28] and used qABC model for numerical optimization problems. If an appropriate similarity measure for the solutions in CABC is defined as in qABC, this useful idea can be incorporated into CABC proposed for combinatorial problems. Details of the qABC algorithm can be found in [28].

In this study we define a similarity measure for TSP solutions to use the new definition for the onlooker bees in CABC algorithm and then call this new CABC as qCABC . Instead of using the Euclidean distance proposed for numerical optimization in qABC, the distance between two solutions x_i *and* x_m is defined by (4) and (5) in TSP,

$$\begin{aligned} dv_{im}(j) &= 0, \quad if\ (x_i[j].afterCity = x_m[j].afterCity\ \ or \\ &\qquad\qquad x_i[j].afterCity = x_m[j].beforeCity) \\ dv_{im}(j) &= 1, \quad else\ not \end{aligned} \tag{4}$$

where dv_{im} is a distance vector for the two closed tours x_i *and* x_m . This definition of similarity is about the same connections of the cities in the closed tours. And the total distance between two solutions, $d(i,m)$ is calculated by using (5)

$$d(i,m) = \sum_{j=1}^{n-1} dv_{im}(j) \quad (5)$$

Steps of qCABC algorithm are follows:

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,…, *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 using their fitness values with (6).

$$p_i = \frac{0.9 \times fit_i}{fit_{best}} + 0.1 \quad (6)$$

   - *Onlooker bees phase:* for each onlooker;
     - Depending on P_i select a solution x_i.
     - Considering the neighborhood radius r, determine the best solution $x_{N_m}^{best}$ among the neighbors of the solution x_i and itself.
     - Produce a new solution $v_{N_m}^{best}$ from $x_{N_m}^{best}$ and evaluate it.
     - Apply the greedy selection process between $x_{N_m}^{best}$ and $v_{N_m}^{best}$.
   - Memorize the best solution achieved yet.
   - Determine the abandoned solution, if exists, replace it with a new randomly produced solution for the scout.
   - $c = c + 1$
7. UNTIL c = *MaxNumber*.

Notice that there is only one difference between CABC and qABC algorithms, which is related to the onlooker bees phase. While, in CABC, employed and onlooker bees determine their new solutions in the same way, in qCABC, they use different definitions.

Since the aim of TSP optimization is minimizing the total closed tour length, the quality of each solution fit_i is calculated by using (7)

$$fit_i = \frac{1}{1 + f(x_i)} \quad (7)$$

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

For the limit value, l is calculated by (8)

$$l = \frac{cs \times D}{3} \quad (8)$$

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

V. SIMULATION RESULTS

In this study, the performance of qABC is evaluated on two benchmark problems and the results are compared with nine different optimization methods in the literature (eight different genetic algorithms having different mutation operators in [29] and CABC algorithm). The first test problem is KroB150. This problem has 150 cities and the optimum tour length of KroB150 is 26130. The second benchmark is KroA200. It contains 200 cities and the optimum tour length for this problem is 29368. These problems and optimum tour lengths can be found in the TSPLIB: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/. Distances between the cities were measured by integer numbers.

For a fair comparison, initial solutions were produced by Nearest Neighbor tour construction heuristic and the same parameter values are taken as in [29]. Considering the searching mechanisms of CABC and qCABC algorithms, the parameter L_{MAX} is set as $n/2$, not Int(sqrt(n)) as in [29]. Table 1 shows the parameter settings of CABC and qCABC algorithms. We carried out 10 runs for each problem with random seeds.

TABLE I. PARAMETER SETTINGS

Parameter	CABC	qCABC
cs	40	40
MaxNumber	20000	20000
P_{RC}	0.5	0.5
P_{CP}	0.8	0.8
P_L	0.2	0.2
L_{MIN}	2	2
L_{MAX}	$n/2$	$n/2$
NL_{MAX}	5	5
r	-	1

The performance comparison of qCABC, CABC and different genetic algorithms is given in Table 2. In order to consider the results presented in [29], in this comparison the best and average percentage error values are used. These percentage errors are calculated by (9)

$$Error(\%) = \frac{L_F - L_O}{L_O} \times 100 \quad (9)$$

where L_F represents the tour length found by the algorithm and L_O is the optimum tour length for the problem.

When the table is examined, the success of CABC and qCABC algorithms can be seen clearly. Especially qCABC algorithm gives the best results in terms of best error for the problems. The convergence graphics of CABC and qCABC are shown in Fig. 1-2 for the first 2000 cycles of the optimization. These figures demonstrate that qCABC is quicker than CABC.

TABLE II. 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	0.3100	**0.6950**	0.4597	0.6034
qCABC	**0.0574**	0.7853	**0.4222**	**0.5145**

VI. CONCLUSIONS

In this paper, a new version of combinatorial ABC algorithm is introduced for solving TSP. The new version is called quick Combinatorial ABC -qCABC algorithm. qCABC is tested on two TSP benchmarks with 150 cities and 200 cities. Experimental results showed that our new algorithm -qCABC- presents good solutions for the considered problems. As future work, we plan to evaluate the performance of new CABC on more test problems and also to analyze the effect of control parameters on the performance of the algorithm.

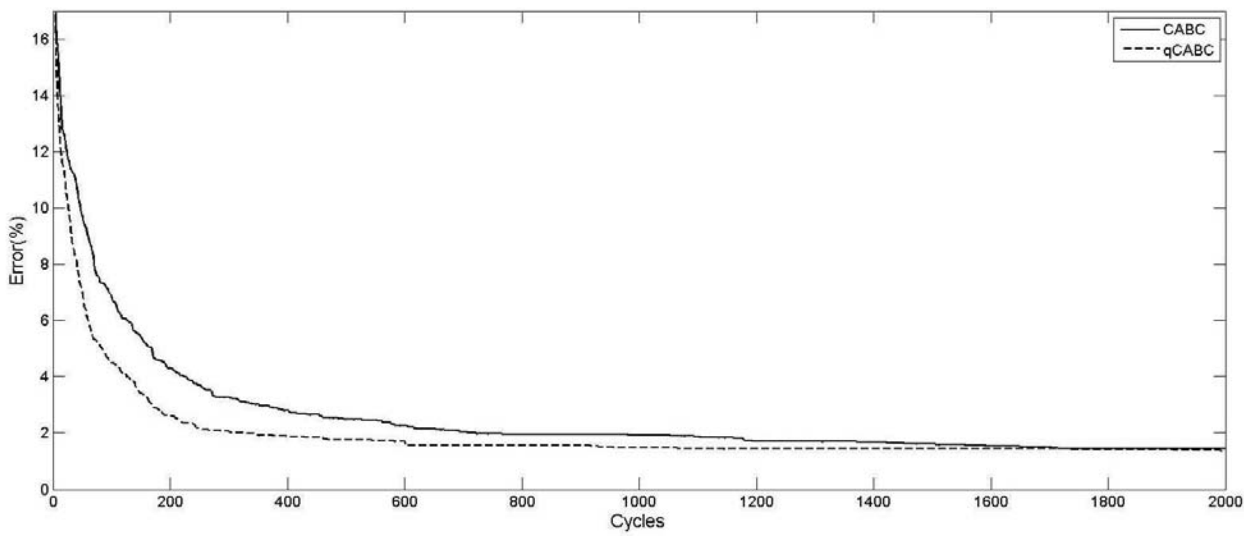

Figure 1. Convergence performance of CABC and qCABC algorithms on KroB150 problem

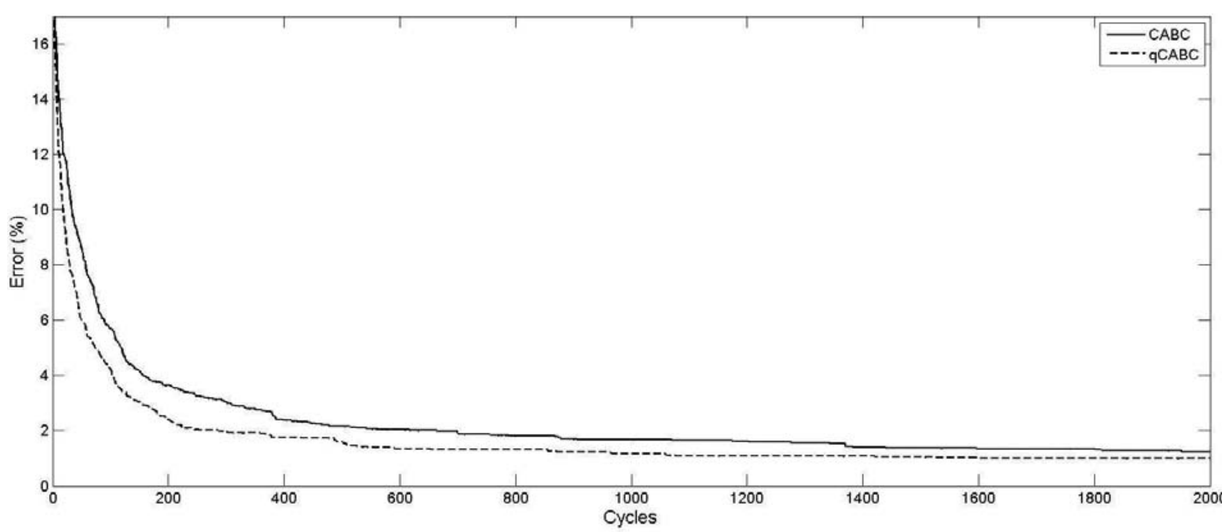

Figure 2. Convergence performance of CABC and qCABC algorithms on KroA200 problem

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] 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.

[5] 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.

[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] M. Tuba and R. Jovanovic, "Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem," International Journal of Computers Communications & Control, vol. 8, no. 3, pp. 477-485, 2013.

[8] 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.

[9] 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.

[10] 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.

[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] 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.

[14] 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.

[15] 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.

[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] Y. Nagata and S. Kobayashi, "A powerful genetic algorithm using edge assembly crossover for the traveling salesman problem," Informs Journal on Computing, vol. 25, no. 4, pp. 346-363, 2013.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," Artif. Intell. Rev., DOI: 10.1007/s10462-012-9328-0, 2012.

[25] L. Li, YR. Chong, LJ. Tan and B. Niu, "A discrete artificial bee colony algorithm for TSP problem," in Proceedings of 7th International Conference on Intelligent Computing (ICIC), 2011, pp. 566-573.

[26] WH. Li, WJ. Li, Y. Yang, HQ. Liao, JL. Li and XP. Zheng, "Artificial bee colony algorithm for traveling salesman problem," in Proceedings of International Conference on Advanced Design and Manufacturing Engineering (ADME 2011), 2011, pp. 2191-2196.

[27] D. Karaboga and B. Gorkemli, "A combinatorial artificial bee colony algorithm for traveling salesman problem," in Proceedings of INISTA 2011: International Symposium on Innovations in Intelligent Systems and Applications, June 2011, pp. 50-53.

[28] D. Karaboga and B. Gorkemli, "A quick artificial bee colony -qABC- algorithm for optimization problems," in Proceedings of INISTA 2012: International Symposium on Innovations in Intelligent Systems and Applications, July 2012, pp. 1-5.

[29] 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.