Jurnal Kejuruteraan 33(4) 2021: 1087-1095
https://doi.org/10.17576/jkukm-2021-33(4)-31

Artificial Bee Colony Algorithm with Proposed Discrete Nearest Neighborhood Algorithm for Discrete Optimization Problems

Amir Masoud Rahimi[a] & Ehsan Ramezani-Khansari[*b]

[a]Department of Civil Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran
[b]Faculty of Civil Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

*Corresponding author: E.R.Khansari@aut.ac.ir

Received 12 March 2021, Received in revised form 01 June 2021
Accepted 02 July 2021, Available online 30 November 2021

ABSTRACT

Travelling salesman problem (TSP) is one the problems of NP-complete family, which means finding shortest complete close tour in the graph. This study seeks to solve this problem using Artificial Bee Colony (ABC) Algorithm along with the proposed Discrete Nearest Neighborhood Algorithm (DNNA). DNNA finds shortest path among points by starting from an arbitrary point. In next steps this links will be a guide to make complete tour. In other words the links in partial tours have higher chance to be in the final solution. In order to improve the final solutions of a single created tour, The employee bees' movement radius has been limited, because of avoidance of long random jump between nodes. To reduce the optimization time of the tours created by the artificial bee colony algorithm, the fixed-radius near neighbor 2-opt algorithm was used as well. In addition, 2 types of scout bee were used for to intensify the probability property of the algorithm. Also, convergence in the probability function of employee bees' movement was prevented by increasing the number of route-creating tours. The first scout bee applies the proposed DNNA and the secondary scout bee improves the partial tours of employee bees in a probable way. Although Althought the average error of proposed ABC algorithm has been 0.371% higher than best solution of all methods, it could improve the solution of 3 problems with average of 3.305%. The proposed algorithm has been better than basic ABC in all tested problems with average of 0.570%.

Keywords: Optimization; artificial bee colony; traveling salesman problem; vehicle routing

NOMENCLATURE

i	Flower number (feature)	P_i	Probability of choosing the flower i in the next move
l_i	Distance between bee and flower i	R	Predetermined movement
l_{Farest}	Distance between the farthest point and the present point (eliminating previously seen points)	U_j	Total unseen points
$l_{Nearest}$	Distance between the nearest point and the present point (eliminating previously seen points)	U_R	Total points in R radius
m	Indicates the employee bee's feature	γ	Coefficient of probability increase of choosing (desired number appropriate to the problem)
Max_{scout}	Length of the longest link made by the scout bee at the beginning of the tour	λ	0 or 1 indicates whether the point is seen by the scout bee in the previous tour or not
Max_{global}	Length of the biggest link in the shortest tour so far		

INTRODUCTION

Depending on its interpersonal interaction, honeybee's colony has been able to tackle crucial issues, such as foraging. Therefore, the swarm intelligence of honeybees has inspired researchers in unraveling complicated issues. Yonezawa and Kikuchi have studied the foraging behavior in bee's colony and established the principles of swarm intelligence (Yonezawa & Kikuchi 1996). Schmickl evaluated honeybees foraging ability using modeling. He concluded that this behavior is adaptive to different conditions and always guides the colony towards an optimal solution (Schmickl, Thenius & Crailsheim 2005). Jakob compared ants and bees algorithms, in which the pheromone factor is removed. Based on his experiments, both algorithms are equally capable in guiding toward the right tour (Vesterstrom & Thomsen 2004). T. Sato and M. Hagiwara offered the genetic algorithm improved by foraging behavior of honey bees (Sato & Hagiwara 1998; Yonezawa & Kikuchi 1996). Karaboga offered the simulation of the foraging behavior algorithm to solve optimization problems, named artificial bee colony algorithm. In this algorithm, he organized the foraging process of honeybees more efficiently. As a result, the bees were categorized into three types of worker, onlooker and scout. Inspired by this collective behavior (Karaboga 2005). Pham et al. offered an algorithm, i.e. bee algorithm which was applied in two and six-dimensional optimization, including near-optimal results (Pham et al. 2011). Lucic and Teodorovic tried solving samples of Travelling Salesman Problem using bee colony algorithm. The results of this study are indicative of bee colony algorithm's capability in solving engineering and NP-Hard problems (Lucic & Teodorovic 2001). Teodorovic et al. continued to apply this algorithm in solving different engineering and management problems successfully (Teodorovic, Lucic, Markovic & Dell'Orco 2006). Teodorovic and Dell'Orco offered the optimization of bee colony, based on a study by Lucic (Dušan Teodorovic & Dell'Orco 2005). Nakrani and Tovey offered the bee algorithm for Dynamic Allocation in an Internet Server Colony. Based on the studies, this algorithm outperforms greedy or static algorithms (Nakrani & Tovey 2003). Wedde et al. improved the bee colony algorithm by introducing the concept of foraging areas, and applied this algorithm, i.e. Beehive Algorithm, to internet connection (Wedde, Farooq & Zhang 2004). Bianco made use of bee colony algorithm in large-scale navigation (Bianco 2004). Chong et al. applied the bee colony foraging algorithm to solve the Job Shop Scheduling problem (Chong, Sivakumar, Low & Gay 2006). Navrat introduced a new method of searching the internet using bee colony algorithm (Návrat 2006). Quijano and Passino exploited the foraging behavior of honey bees to solve the resource allocation problem (Quijano & Passino 2010). Karaboga and Basturk improved the bee colony algorithm and applied it in many problems. As a result, they found out that this algorithm is capable of escaping local optima (Karaboga & Basturk 2007, 2008). Baykasoùlu et al. carried out a study on the algorithms inspired by the foraging behavior of honey bees and drew on the bee colony algorithm to solve generalized assignment problem (Baykasoğlu, Ozbakir & Tapkan 2007). Karaboga and Akay made a few corrections to the bee colony algorithm (Karaboga & Akay 2011). Aghazadeh and Meybodi applied the bee algorithm along with Learning Automaton (Aghazadeh & Meybodi 2011). Wong et al. tried solving the Travelling Salesman problem (TSP) using bee colony algorithm along with Fragmentation State Transition Rule (FSTR) (Wong, Low, Chong et al. 2009). Travelling salesman problem (TSP) is one the problems of NP-complete family, which means finding shortest complete close tour in the graph. Bhagade and Puranik made use of the bee colony algorithm to solve car navigation problem. In this study, the Nearest Neighbor algorithm, along with bee colony, was used (Bhagade & Puranik 2012). Ji and Wu tried to solve car navigation problem with Capacitated Vehicle Routing Problem with Time-dependent travel times condition (Ji & Wu 2011). Many types of TSP has been tried to solve by variety of methods during recent decades (Laporte, 2009). Many other researchers utilized bee colony as an acceptable meta-heuristic method for optimization in engineering problems (Gholipour, Khosravi & Mojallali 2013; Jiang, Yue, Li, Wang & Guo 2015; Tolabi, Moradib & Tolabia 2013; Toloei, Zarchi & Attaran 2017).

Althought agent-based programming is harder and more complex than structured programming, it may improve the result of the algorithm and process time. Sahin et al. (2021) tested two multi-criteria and combinatorial ABC algorithms using mutation and crossover operators test suite that maximizes a fitness function combining various goals for object-oriented software. The experimental results reveal that introducing an archive into the ABC algorithm provides fast convergence compared to the basic ABC.

Kiran (2021) replaced the rule of solution update with XOR gate in binABC algorithm to solve the discretely-structured solution space. The rest of components in binABC are the same as with the basic ABC algorithm. The proposed algorithm is first compared with the basic ABC and binABC on CEC2015 set. The comparision with basic ABC showed that it can be effective in solving binary optimization problems.

Wang and Szeto (Wang & Szeto 2021) tried to minimize the carbon dioxide emissions of the repositioning vehicle besides achieving the target inventory level at stations as much as possible within the time budget. Two methods, namely heuristic and exact methods, are proposed and incorporated into the enhanced ABC algorithm to compute the loading/unloading quantities at each stop. The results showed that the algorithm can provide optimal for small instances.

In the present study, it is attempted to find a more suitable solution for TSP, making some alterations in the bee colony foraging algorithm.

THE PROPOSED ARTIFICIAL BEE COLONY ALGORITHM

The general pseudo code of bee colony is as follows:

1. The first scout bee creates a partial tour using proposed Discrete Nearest Neighborhood Algorithm (DNNA).
2. The partial tour is saved in the hive and transferred to onlooker and employee bees.
3. The onlooker and employee bees, committed to the movement radius, complete the tour.

   a. The employee bees with inefficient tours change to secondary scout bee.
4. The completed tour is optimized by fixed-radius near neighbor 2-opt (FRRN 2-opt)
5. The completed tour is examined

THE FIRST SCOUT BEE AND CREATING AN INCOMPLETE TOUR

The purpose of the first scout bee is creatingpartial tours. In the proposed algorithm, initially, the hive is placed outside of the graph of points (flowers). Now a scout bee leaves the hive and chooses a flower out of graph completely randomly and uses the DNNA to choose the next point, using the Nearest Neighborhood Algorithm (NNA) to move among the points, no further than where the nearest neighbor was previously seen. At this time, the first scout bee flies randomly to a point not seen before and continues the NNA. This algorithm is called discrete probable since the links are always unconnected and yields different random results depending on the chosen point, and does not form a close loop as well. The pseudo code of the first scout bees' movement using proposed DNNA (Figure 1).

Generally, the scout bee's flying pattern in the proposed algorithm can be explained in two phases: random movements only to choose the next point to start the NNA, and movements made by using NNA in order to make a link between the points. The moving process of a first scout bee is illustrated schematically in the figure 2. Figure 3 depicts the flowcahrt of proposed scout bee movements.

```
N=Node(flower) , i=Node number : {1,2,…,i}
Ns, Nj, Seen_point = {}
** Random select  Ns : Ns ϵ Ni – seen_point
while Ns ≠ {}
        Nj←nearest node to Ns
        ifNj ϵ Seen_point
                thengoto **
        elseNj ∉ Seen_point
                thenNj←Seen_point
                Ns=Nj
        end
end
```

FIGURE 1. DNNA pseudo code

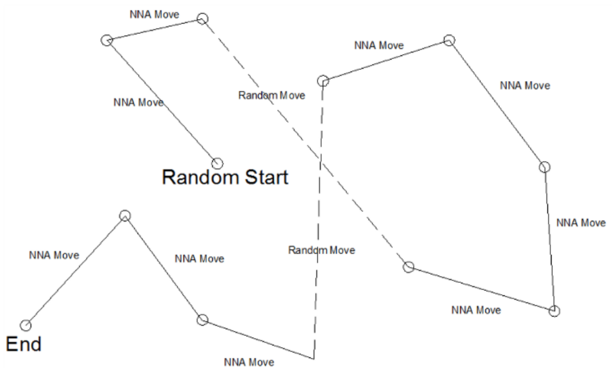

FIGURE 2. Procedure of first scout bee

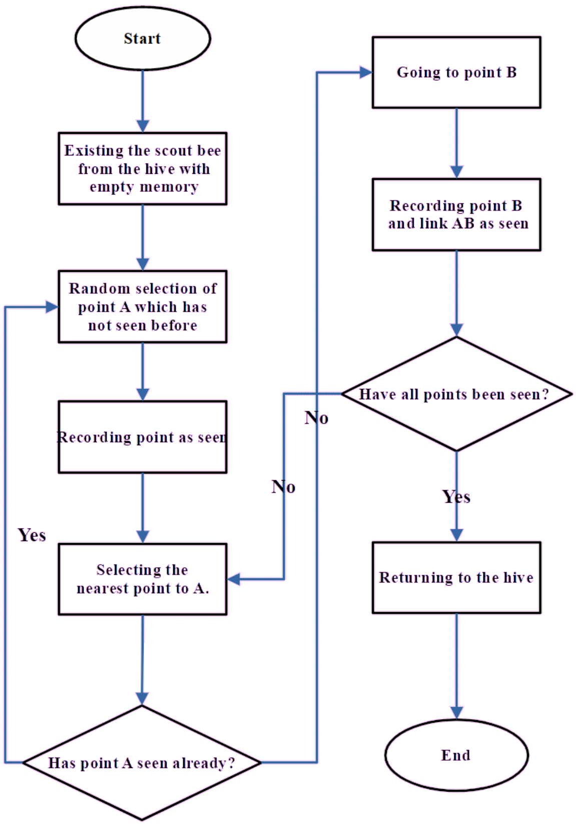

FIGURE 3 Scout bee movement flowchart

THE EMPLOYEE AND ONLOOKER BEE COMMITTED TO MOVEMENT RADIUS TO COMPLETE THE TOUR

At this phase, all the links created by the first scout bee is transferred to the employee bee and the beehive is located on a flower randomly. Based on the probability function, the employee bees develop a tour (equation 1, 2 & 3).

$$P_i = \frac{S_i}{\sum_{i=1}^{n} S_i} \quad (1)$$

$$S_i = \frac{1}{l_i}(\gamma^{\Psi}) \quad (2)$$

$$l_{norm} = \frac{l_{farest} - l_i}{l_{farest} - l_{nearest}} \quad (3)$$

i: flower number (feature)
P_i: probability of choosing the flower i in the next move
l_i: distance between bee and flower i
γ: coefficient of probability increase of choosing (desired number appropriate to the problem)
l_{Farest}: distance between the farthest point and the present point (eliminating previously seen points)
$l_{Nearest}$: distance between the nearest point and the current point (eliminating previously seen points)
λ: 0 or 1 indicates whether the point is seen by the scout bee in the previous tour or not.

$$R = Max\begin{Bmatrix} Max_{scout} \\ Max_{global} \end{Bmatrix} \quad (1)$$

$$i = \begin{Bmatrix} i \in U_R & ,U_R \neq 0 \\ i \in U_j & ,U_R = 0 \end{Bmatrix} \quad (2)$$

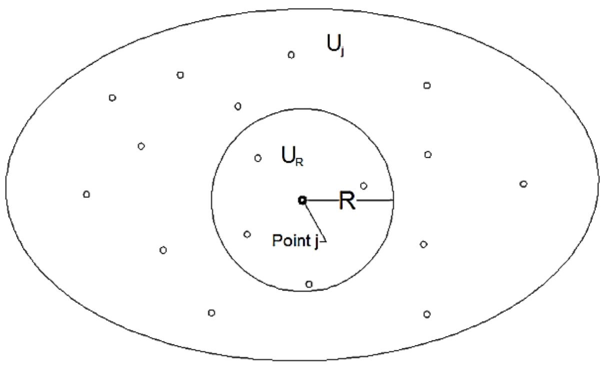

FIGURE 4. Regions of employee bee movement

As mentioned, λ can be 0 or 1. λ indicates the effect of not seeing the examined link by scout bee (or the previously seen link in the best and last tour which is applicable in the second tour onward). In this function, λ is a coefficient with an optional value (determined by the proposed algorithm user). This coefficient determines the impact of information saved in the employee bee's memory (received from the previous tour or the first scout bee) in the proposed algorithm. In many probable functions in metaheuristic methods, λ coefficient is a number increased by rising the repetition cycles and yields convergence to one answer.

Here, however, our purpose is to prevent global convergence. As a result, after finding the possible tour, λ coefficient is assumed fixed and not influenced by the number of tours. Therefore, the search is entirely random and does not lead to a solution.

The coefficient of a link being saved in employee's memory is considered $\Psi = l_{norm}^{\lambda}$ which can have three values in l_i equation: $l_i = l_{farest}$, $l_i = l_{nearest}$, $l_i \neq l_{farest} \& l_{nearest}$.

In $l_i = l_{nearest}$, the numerator is zero, and as a result, Ψ coefficient (effect of previous information) would be zero as well. In other words, although this link is seen once (λ=1), no extra value or increased probability is attributed to it, due to previous observation (Ψ=1). In the second situation, in which li=lnearest, the numerator and denominator , as seen before, are equal. Therefore, γ=Ψ which indicates that the next flower, whose information is being estimated, is the nearest one and seen previously as well. In the third situation, which a value between the above 2 cases is chosen, 1< Ψ< γ. It can be concluded that the proposed Ψ algorithm is under one of the following conditions:

$$l_i = l_{farest} \rightarrow \Psi = 1$$
$$l_i = l_{nearest} \rightarrow \Psi = \gamma$$
$$l_i \neq l_{farest} \,\&\, l_{nearest} \rightarrow 1 < \Psi < \gamma$$

Therefore, a previously seen tour could be finally influenced by a coefficient in the range γ ≤Ψ ≤ 1 (of course assuming λ=1). The coefficient γ is experimental and will be applied by the beehive (user) after repeating for a couple of times. In addition, as the probability of previously seen links in employee's memory increases, the employee bee tends to follow its own memory, rather than the tour ahead (however, this tendency will be probable as well).

Following each choosing process, the employee bee does not continue its way, but is returned to the hive and introduces the tour it has followed since leaving the hive. The onlooker bees, whose number is chosen as much as the employee bees, examine the tours and rank them based on equation 3.

$$\delta = \frac{\frac{1}{l_m}}{\sum \frac{1}{l_m}} \quad (4)$$

Index m indicates the employee bee's feature (name) which has proceeded the rout as long as l, and is demonstrating it now. A δ will be dedicated to each employee bee by the hive (δm). This number will be rounded to the nearest number. Next, onlookers will be sent after the employee bee to reach the end of the provided tour created by the employee bee. Figure 5 shows the employee bee movement flowchart.

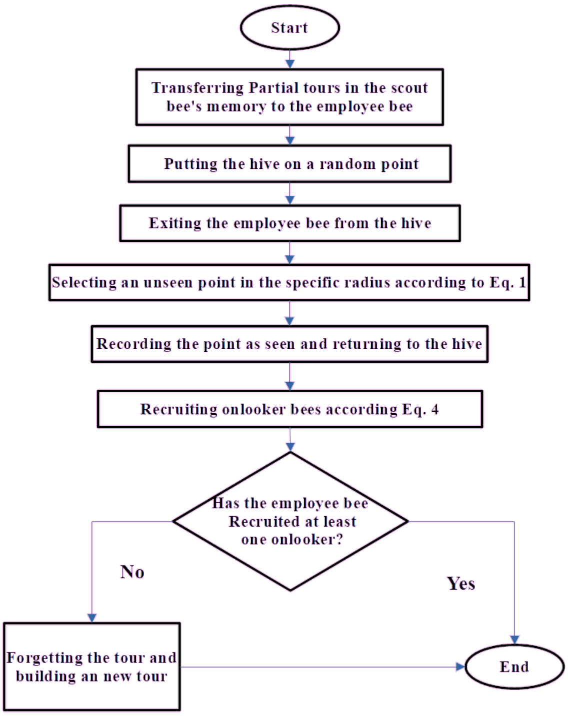

FIGURE 5 The Employee bee movement flowchart

THE SECONDARY EMPLOYEE BEE

The employee bee, if δ is zero, would prevent getting back to the last point and continuing its tour. This time, it will leave the hive as a bee which has just left the hive and will forget all the points and the followed tour seen previously.

This employee bee, which creates part of the tour at once and then examines it, is called the "secondary scout bee". Like other employee bees, the secondary scout bee, on its way back to the hive, publicizes its tour and starts recruiting.

THE 2-OPT ALGORITHM

At the end of a cycle of the proposed algorithm which was introduced and chosen as the best (and shortest) tour in the hive, the tour will be optimized by the 2-opt algorithm, which is done here by the 2-opt fixed-radius near neighbor algorithm, introduced by Bentley (Bentley 1992).

Examining both links in a complete tour, this algorithm tries to create a new shorter tour. Simply put, this algorithm prevents the links from crossing each other.

The 2-opt algorithm is really efficient. However, in order to improve a tour in a graph with N points, it is necessary that (N×N) controls the new tour, a time-consuming process. Thus, at the end of each complete cycle the best tour is selected using a revised version of 2-opt algorithm, i.e. FRNN 2-opt, instead of the classic 2-opt algorithm. In this algorithm, the same method as 2-opt is applied to examine the links. While the searching time consumption, one of the classic 2-opt's shortcomings, will be reduced and optimizing the sites of choosing the candidate links to be replaced.

The revised 2-opt method is based on the experimental observation that if we want to make an alteration between the two links resulting in a shorter tour, at least the length of one of the two interchanged links must be reduced after applying 2-opt. in other words, if one of the links is supposed to be candidate as the one to reduce the length of the tour, it must be replaced with the shorter links only. Thus, we must look for the points in a radius as long as the candidate link that would be shorter whenever the intended link was eliminated and linked to the point. As demonstrated in the image, searching around D point and examining only the three DC, DB and EB links has resulted in a shorter tour (Figure 6).

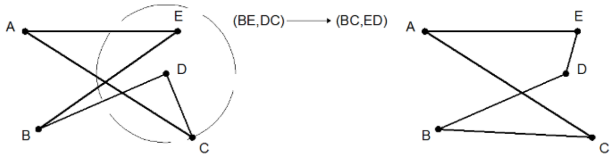

FIGURE 6.Procedure of edge change in FRNN 2-opt algorithm

Since the employee bees' moving pattern is pre-determined here, no multiple and complicated crossing will occur. As a result, the FRNN 2-opt algorithm, however less capable, has been able to optimize the tour efficiently (Wong, Low & Chong 2009).

GLOBAL SEARCHING AND THE HIVE'S MEMORY

In order to conduct a global search, the best tour of the previous cycles will be transferred to the next. Thus, the search will be conducted around the best solution, and a better global solution will be build. In other words, the local search is always carried out around the global search, and getting stuck in local optima is prevented by revising the links provided by the first scout bee at the beginning of each cycle producing new unconnected links.

RESULTS AND DISCUSSION

The proposed algorithm was applied on sample travelling salesman problems by repeating the cycle for 1500 times and with γ=2 parameter. For this purpose, MATLAB software and a computer with Intel CORE ™ i7 processor and 6GB RAM was used.

Table 1 compares the solution of the proposed ABC algorithm with the the best solution of all methods and the basic ABC algorithm (Lucic & Teodorovic 2001). Comparison with the basic ABD algorithm shows whether the proposed algorithm could improve it or not.

Comparison with the best solution also shows how capable the proposed ABC algorithm is in solving the TSP problem. Althought the error of proposed ABC algorithm has been 0.371% higher than best solution of all methods, it could improve the solution of 3 problems with average of 3.305%. The proposed algorithm has been better than basic ABC in all tested problems with average of 0.570%. It can be seen the error percentage of proposed ABC algorithm has not been dependent on the number of points of TSPs, thus it can be inferred that the random parts of the proposed algothim would functioned well. The last column of table 1 state the amount of processing effort which is needed for solving each problem. As it shows the relation between problem complexity and process time is exponentional, which is the nature of NP-family problems and shows the procedure of proposed algorithm can be logical.

Table 2 shows the comparison between propose algorithm and other well-known heuristic algorithms. The best solutions are in bold type. It shows that the solutions of proposed algorithm in the tested problems were better than other thee heuristic algoritms. It should be mentioned other heuristic algorithm was applied in the basic form.

This research aimed to improve the basic ABC algorithm in two ways. On the one hand, the proposed algorithm attempts to build a better initial solution so that it can be more easily optimized in later steps. It is a common method for improving heuristic algorithms.

On the other hand, the initial route is made by building partial tours and connecting them to each other. These partial tours are optimum for some points.

Furthermore, the long jumps were avoided by using a spesific radius which influence the probabilty of selection of farther points.

TABLE 1. Comparing the results of proposed ABC algorithm and best solution

Problem	The best value achieved by the proposed ABC algorithm (P)	The best solution of all methods (A)	Error percentage comparing with the best of all methods	The best value achieved by ABC algorithm (B)	Error percentage comparing with the ABC algorithm	Average CPU process time (Second)
bayg29	9074.1000	9074.148	-0.001	-	-	344
att48	33523.7085	33523.709	0.000	-	-	786
Eil51	428.8718	426.000	0.674	-	-	1189
Berlin52	7544.3659	7542.000	0.031	7544.366	0.00	1200
St70	677.1096	675.000	0.313	678.621	-0.223	2967
Pr76	108280.4566	108159.000	0.112	108790	-0.471	3477
Eil76	555.7688	545.388	1.903	-	-	4896
gr96	512.6912	512.309	0.075	-	-	7358
Kroa100	21311.0000	21282.000	0.136	21441.5	-0.612	8468
kroaC100	20880.2012	20750.763	0.624	-	-	10538
lin105	14529.5632	14382.996	1.019	-	-	12687
gr120	1649.3278	1666.509	-1.031	-	-	28617
ch130	6199.4062	6110.861	1.449	-	-	426973
ch150	6659.0000	6528.000	2.007	-	-	58621
gr202	501.1347	549.998	-8.884	-	-	97530
Tsp225	4044.7201	3916.000	3.287	4065.56	-0.515	117737
A280	2697.4771	2579.000	4.594	2740.63	-1.600	253906
Average (%)	0.371	-	-0.570			

TABLE 2. Comparing the results of proposed ABC algorithm and other algorithms

Algorithm / Problem	Proposed ABC algorithm (Iteration=1500)	Ant Colony Optimization (ACO) (Iteration=1500)	Genetic Algorithm (GA) (Iteration)	Simulated Annealing (SA) (iteration=2000)
bayg29	**9074.1000**	9685.3581	9074.1480 (1243)	9151.2852
att48	**33523.7085**	40884.4343	33784.0270 (2489)	34498.3812
Eil51	**428.8718**	576.779	440.0559 (4031)	442.1317
Berlin52	**7544.3659**	9640.0343	7739.6099 (3337)	7967.8442
St70	**677.1096**	939.1902	682.6597 (4681)	710.4497
Pr76	**108280.4566**	123561.3258	116337.6020 (5091)	116549.1685
Eil76	555.7688	599.3587	**554.7059 (9034)**	580.6342
gr96	**512.6912**	545.3682	527.2520 (6329)	539.228
Kroa100	**21311.0000**	23698.1290	22197.2059 (5210)	22357.1772
kroaC100	**20880.2012**	21983.3698	21752.3650 (9035)	21677.3694
lin105	**14529.5632**	15684.3987	14880.6031 (9547)	15111.2439
gr120	**1649.3278**	1698.3579	1753.4668 (9677)	1683.1326
ch130	**6199.4062**	6821.2679	6765.6444 (9678)	6344.4263
ch150	**6659.0000**	7561.3192	7606.6787 (9515)	7030.1595
gr202	**501.1347**	532.6872	546.6265 (9990)	517.2730
Tsp225	**4044.7201**	7235.2587	69211.5141 (9792)	6827.9882
A280	**2697.4771**	2932.5781	2817.7865 (3780)	2936.0204

CONCLUSION

Based on the results, revising the bee colony algorithm provides better solutions to the travelling salesman problem. As a result of applying DNNA algorithm, ABC algorithm started the solving process with a better solution. It was observed that DNNA has created edges as much as at least 50% of the points. 75% of these edges were observed in final optimized tours, indicating how valuable the created edges by DNNA are. Simultaneously applying the limiting strategy to the movement radius of employee bees, which has reduced and simplified the crossing of edges, and 2-opt in FRNN algorithm instead of 2-opt algorithm, has reduced the time consumption and reserved optimization quality of created tour by ABC algorithm. Applying the DNNA algorithm at the beginning of each cycle of creating tour, has helped preventing from getting stuck in local optimization and analyzing other possible optimizations.

ACKNOWLEGMENT

The authors would like to thank the Government Engineering College for supporting this research.

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Aghazadeh, F., & Meybodi, M. R. 2011. Learning bees algorithm for optimization. *International Conference on Information and Intelligent Computing* 18: 115–122.

Baykasoğlu, a, Ozbakir, L., & Tapkan, P. 2007. Artificial bee colony algorithm and its application to generalized assignment problem. *Swarm Intelligence: Focus on …* 532(December): 113–144. http://www.intechopen.com/source/pdfs/497/InTech-Artificial_bee_colony_algorithm_and_its_application_to_generalized_assignment_problem.pdf

Bentley, J. J. 1992. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing* 4(4): 387–411.

Bhagade, A. S., & Puranik, P. V. 2012. Artificial bee colony (ABC) algorithm for vehicle routing optimization problem. *International Journal of Soft Computing and Engineering* 2(2): 329–333.

Bianco, G. M. 2004. Getting inspired from bees to perform large scale visual precise navigation. Intelligent Robots and Systems, 2004.(IROS 2004). *Proceedings. 2004 IEEE/RSJ International Conference* On 1: 619–624. IEEE.

Chong, C. S., Sivakumar, A. I., Low, M. Y. H., & Gay, K. L. 2006. A bee colony optimization algorithm to job shop scheduling. *Proceedings of the 38th Conference on Winter Simulation*, 1954–1961. Winter Simulation Conference.

Gholipour, R., Khosravi, A., & Mojallali, H. 2013. Parameter estimation of lorenz chaotic dynamic system using bees algorithm. *International Journal of Engineering* 26(3): 257–262.

Ji, P., & Wu, Y. 2011. An improved artificial bee colony algorithm for the capacitated vehicle routing problem with time-dependent travel times. *Tenth International Symposium on Operations Research and Its Applications (ISORA 2011):* 75–82.

Jiang, D., Yue, X., Li, K., Wang, S. & Guo, Z. 2015. Elite opposition-based artificial bee colony algorithm for global optimization. *International Journal of Engineering* 28(9): 1268–1275.

Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Karaboga, D., & Akay, B. 2011. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing* 11(3): 3021–3031.

Karaboga, D., & Basturk, B. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization* 39(3): 459–471.

Karaboga, D., & Basturk, B. 2008. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing* 8(1): 687–697.

Kiran, M. S. 2021. A binary artificial bee colony algorithm and its performance assessment. *Expert Systems with Applications* 175: 114817. https://doi.org/https://doi.org/10.1016/j.eswa.2021.114817

Laporte, G. 2009. Fifty years of vehicle routing. *Transportation Science* 43(4): 408–416.

Lucic, P., & Teodorovic, D. 2001. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. *Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis:* 441–445.

Nakrani, S. & Tovey, C. 2003. On honey bees and dynamic allocation in an internet server colony. *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects,* 1–8. Citeseer.

Návrat, P. 2006. Bee hive metaphor for web search. *Communication and Cognition-Artificial Intelligence* 23(1–4): 15–20.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. 2011. The bees algorithm-A novel tool for complex optimisation. Intelligent Production Machines and Systems-2nd I* *PROMS Virtual International Conference (3-14 July 2006)*. sn.

Quijano, N., & Passino, K. M. 2010. Honey bee social foraging algorithms for resource allocation: Theory and application. *Engineering Applications of Artificial Intelligence* 23(6): 845–861.

Sahin, O., Akay, B. & Karaboga, D. 2021. Archive-based multi-criteria Artificial Bee Colony algorithm for whole test suite generation. *Engineering Science and Technology, an International Journal* 24(3): 806–817. https://doi.org/https://doi.org/10.1016/j.jestch.2020.12.011

Sato, T., & Hagiwara, M. 1998. Bee system: finding solution by a concentrated search. *IEEJ Transactions on Electronics, Information and Systems* 118(5): 721–726.

Schmickl, T., Thenius, R., & Crailsheim, K. 2005. Simulating swarm intelligence in honey bees: foraging in differently fluctuating environments. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, 273–274. ACM.

Teodorovic, Dušan, & Dell'Orco, M. 2005. Bee colony optimization–a cooperative learning approach to complex transportation problems. *Advanced OR and AI Methods in Transportation,* 51–60.

Teodorovic, Dusan, Lucic, P., Markovic, G. & Dell'Orco, M. 2006. Bee colony optimization: principles and applications. *Neural Network Applications in Electrical Engineering,* 2006. NEUREL 2006. 8th Seminar On, 151–156. IEEE.

Tolabi, H. B., Moradib, M. & Tolabia, F. B. 2013. New technique for global solar radiation forecast using bees algorithm. *International Journal of Engineering* 26(11).

Toloei, A., Zarchi, M., & Attaran, B. 2017. Numerical survey of vibrational model for third aircraft based on hr suspension system actuator using two bee algorithm objective functions. *IJE TRANSACTIONS C: Aspects* 30(6): 887–894.

Vesterstrom, J., & Thomsen, R. 2004. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. *Evolutionary Computation, 2004. CEC2004. Congress On,* 2, 1980–1987. IEEE.

Wang, Y., & Szeto, W. Y. 2021. An enhanced artificial bee colony algorithm for the green bike repositioning problem with broken bikes. *Transportation Research Part C: Emerging Technologies* 125: 102895. https://doi.org/https://doi.org/10.1016/j.trc.2020.102895

Wedde, H. F., Farooq, M. & Zhang, Y. 2004. BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. *International Workshop on Ant Colony Optimization and Swarm Intelligence,* 83–94. Springer.

Wong, L.-P., Low, M. Y. H. & Chong, C. S. 2009. An efficient bee colony optimization algorithm for traveling salesman problem using frequency-based pruning. *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference On,* 775–782. IEEE.

Wong, L.-P., Low, M. Y. H., Chong, C. S., Pham, D. T., Eldukhri, E. E. & Soroka, A. J. 2009. A bee colony optimization algorithm with the fragmentation state transition rule for traveling salesman problem. *Proceedings of the 2009 Conference on Innovative Production Machines and Systems (IPROMS 2009),* 399–404. Cardiff University, Cardiff, UK.

Yonezawa, Y. & Kikuchi, T. 1996. Ecological algorithm for optimal ordering used by collective honey bee behavior. *Micro Machine and Human Science, 1996., Proceedings of the Seventh International Symposium,* 249–256. IEEE.