Increased Efficiency of the Artificial Bee Colony Algorithm Using the Pheromone Technique

Abdul Hadi Alaidi
Programming Department, Computer Science and Information Technology College, Wasit University
Wasit, Iraq
alaidi@uowasit.edu.iq

Soong Der Chen
College of Computing and Informatics
Universiti Tenaga Nasional
Selangor, Malaysia
chensoong@uniten.edu.my

Yeng Weng Leong
College of Computing and Informatics
Universiti Tenaga Nasional
Selangor, Malaysia
ywleong@uniten.edu.my

Received: 1 September 2022 | Revised: 15 September 2022 | Accepted: 17 September 2022

Abstract-Artificial Bee Colony (ABC) is a powerful metaheuristic algorithm inspired by the behavior of a honey bee swarm. ABC suffers from poor exploitation and, in some cases, poor exploration. Ant Colony Optimization (ACO) is another metaheuristic algorithm that uses pheromones as a guide for an ant to find its way. This study used a pheromone technique from ACO on ABC to enhance its exploration and exploitation. The performance of the proposed method was verified through twenty instances from TSPLIB. The results were compared with the original ABC method and showed that the proposed method leverages the performance of ABC.

Keywords-artificial bee colony; pheromone; ant colony optimization

I. INTRODUCTION

Artificial Bee Colony (ABC) is a well-known algorithm for solving continuous function optimization problems. Although ABC is more effective than other biologically inspired algorithms, it suffers from a slow convergence rate in large search spaces [1-3]. In addition, one of the problems of ABC is the blind search strategy for food sources. In nature, honey bees interact with each other primarily through chemistry-driven behavioral communication. However, in the conventional ABC algorithm, honeybees only dance to share information. Therefore, it is necessary to incorporate chemistry communication in addition to behavioral to comply with the reality in nature and the basic principles of ABC. The attraction pheromone and transition strategy were used in [3, 4] as a communication strategy to share information between bees and solve the blindness search. It was assumed that each bee moves to a solution and deposits the attraction pheromone that attracts other bees to the object where the pheromone strength is higher than the others. This mechanism enabled bees to search for food more effectively. Other studies used pheromones to overcome the limitation of ABC exploitation by making bees communicate and share their findings [3, 4]. Scout bees use pheromones as a guide when looking for a new food source, and pheromones are updated at all stages. Moreover, the amount of deposit pheromones is a function of fitness measures.

ABC integrates grenade explosion and the Cauchy operator to avoid random searches and solve Dynamic Economic Emission Dispatch (DEED) problems [5]. In [6, 7], methods were used for routing and improving the performance of Wireless Sensor Networks. The memory mechanism plays a crucial role in the guide of the search process by memorizing valuable information and exploring new search spaces. In [8], the memory mechanism was used to improve ABC performance, addressing the fundamental problem of ABC, which is the exploration and late convergence. This study aimed to solve the problem of exploration and late convergence in ABC using Random Memory (RM) and Elite Memory (EM), leading to a more desirable version of ABC called ABCWOA. In the ABCWOA algorithm, RM uses the bait search stage in the Whale Optimization Algorithm (WOA). At the same time, EM is dynamically controlled based on the number of iterations and used to increase the convergence rate. Furthermore, ten standard data sets from the UCI machine learning repository were used for the evaluation, showing that ABCWOA had an improved convergence rate and less noise.

ABC is used to solve many types of problems. For example, an unrelated parallel machine scheduling problem was solved with ABC using a multi-sub-colony [9]. In addition, a multi-strategy evolutionary learning ABC algorithm was proposed to solve the path problem of an unmanned autonomous helicopter [10]. This study aimed to increase the performance of ABC using pheromone as a guide for bees and to overcome the ABC exploitation and exploration problem.

Corresponding author: n Abdul Hadi Alaidi

II. THE ARTIFICIAL BEE COLONY

The ABC is a well-studied algorithm developed to solve continuous function optimization problems [11], simulating the foraging behaviors of a swarm of bees to find food. ABC is used successfully in many fields including energy management in mobile devices [12] and IoT routing [13]. ABC is divided into three groups: employed, onlookers, and scout bees. At first, employed bees search for food sources and share information with an onlooker. The onlooker bee tries to search for a better food source in the neighborhood using the knowledge passed from the employed bees. If the food source is abandoned to escape the limitation of food enhancement, the bees employed for that food source become scouts and randomly search for a new food source. In ABC, employed and onlooker bees perform the exploitation, while scout bees control the exploration process. The employed bees use (1) to search for neighbors by moving from an old position x_{ij} to the new position v_{ij}, if and only if the new position's fitness is better than the old position's.

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \quad (1)$$

where x_i is the old position and v_i is the new position of the i^{th} employed bee, φ_{ij} is a random real number in the range [-1, 1], j is an arbitrary integer number between 1 and the problem's dimension, and k is a random number between 1 and the number of employed bees. The onlooker bee selects a food source and evaluates all employed bees depending on the probability:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (2)$$

where fit is the fitness value of the solution of the employed bee:

$$fit_i = \begin{cases} \frac{1}{1+f(x_i)}, & f(x_i) \geq 0 \\ 1 + |f(x_i)|, & f(x_i) < 0 \end{cases} \quad (3)$$

where $f(x_i)$ stands for the value of the objective function to be optimized. Like the employed bee, the onlooker bee uses (1) to update its position and explore the selected food source. When a food source cannot be further enhanced within a pre-defined trial limit, it is abandoned and the corresponding employed bee becomes a scout and randomly searches for a new food source. The combination of exploration and exploitation search in ABC is important as it enables the algorithm to search the solution space for high-quality solutions and prevents them from getting stuck in local optima. In ABC, the employed bees maintain the current solution while the onlooker bees allow the best solution to be exploited. Furthermore, scout bees try to remove stagnating solutions and explore new ones.

```
Initialization Phase
REPEAT
    Employed Bees Phase
    Onlooker Bees Phase
    Scout Bees Phase
    Memorize the best solution achieved so far
UNTIL(Cycle=Maximum Cycle Number)
```

Fig. 1. The standard ABC.

III. ANT COLONY OPTIMIZATION

Another non-deterministic algorithm used to solve an optimization problem is Ant Colony Optimization (ACO). ACO was inspired by ants seeking the shortest route between a source of food and their colonies [14]. ACO seeks the optimal solution for the computation problem by building one solution and adding it to solution components until the wanted solution is built. ACO has been successfully applied to solve the traveling salesman problem. However, as the complexity of the problem increases the accuracy of the ACO algorithm decreases. In nature, ants use pheromones as indirect communication between them. In ACO, pheromones are represented by numeric values modified by every solution component. These values influence the movement of ants in the next iteration in a way that edges between nodes with large values (high levels of pheromones) are more likely to be visited. Furthermore, the ACO model allows for the inclusion of heuristic information to guide the ants to complete a feasible solution.

```
Set parameters, initialize pheromone trails
while (condition is not satisfied) do
    ConstructAntSolutions
    DaemonActions {optional}
    UpdatePheromones
End While
```

Fig. 2. The ACO algorithm.

Each ant starts from a random location and makes a trip. When being in location i at the t^{th} iteration, ant k applies a probabilistic action before it is transmitted to the next unvisited location j. Finding the probability for ant k is given by [15]:

$$P_{ij}^{k}(t) = \frac{[T_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[T_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}}, \; if\, j \in N_i^k \quad (4)$$

where $\eta = 1/d_{ij}$ is the value of the heuristic available in advance, a and β are two parameters that define the relative influence of the pheromone trail and the heuristic information, and N_i^k is the feasible neighborhood of the ant k. Parameters α and β play a main role in choosing the next location to be visited. If $\alpha = 0$, there is no pheromone impact on the movement of the ants. However, if $\beta = 0$, only pheromone strengthening works. It has been suggested that all ants construct the same path consequently and a trade-off exists between the pheromone trail and the impact of the heuristic information [15].

The pheromone update procedure is performed in two steps. First, the amount of pheromone on the edge that joins two locations is decreased by a constant factor. The next step is to put a certain amount of pheromone on that particular edge. The update of the pheromone is calculated by:

$$T_{ij}(t+1) = (1-\rho).T_{ij}(t) + \sum_{K=1}^{M} \Delta T_{ij}^{K}(t) \quad (5)$$

where ρ is greater than 0 and equal to or less than 1. The parameter ρ is used to avoid accumulating a high level of the pheromone trail by lowering the pheromone level. Similarly, it helps ant to discard bad decisions early taken and discover a new solution. Moreover, $\Delta T_{ij}^{k}(t)$ represents the quantity of pheromone that will be added, defined as:

$$\Delta T_{ij}^{K}(t) = \begin{cases} 1/L_{K(t)} & \text{If edge } (i,j) \text{ is used by the ant} \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

where $L_{K(t)}$ is the length of the K^{th} ant's trip. Equation (6) shows that adding more pheromones to an edge gives a better ant trip, and edges with high pheromones are more likely to be visited in the next iteration. In AS, all ants add a certain quantity of pheromone on the visited edge. This method does not provide additional support for the best solution found, although letting all ants update the pheromone trails may help the algorithm avoid an early stagnation and move to different search space.

IV. Using Pheromones with ABC to Solve TSP

The Traveling Salesman Problem (TSP) was used as a test case for the proposed algorithm. TSP is an easy to describe well-known problem, but hard to solve. The purpose of TSP is to find the shortest path between groups of cities with minimum cost.

A. Initialization Phase

Pheromones can integrate into ABC to give memory-like behavior. The proposed algorithm initializes the parameters and then initializes the pheromones using the neighbor search algorithm to find the start path to update the pheromone using (5). Finally, the food source (solutions) is generated using (4).

```
    Initialize maxIteration, popSize, limit, Beta &
initialize pheromone trails
    for i = 1 to popSize/2
        foodSource[i] = InitializeUsingPheromone()
        unsuccessfulCycles[i] = 0
        fitness[i] = CalculateFitness(foodSource[i])
    end for
```

Fig. 3. Initialization.

B. Employed Bee Phase

The employee bee is responsible for searching in the neighborhood for new solutions and changing positions if the new position is better than the old one. Figure 4 shows the employee bee phase algorithm.

```
for i = 1 to popSize/2
    newSol = neighborSeach(foodSource[i], i)
    newCost = CalculateFitness(newSol)
    if newCost < fitness[i] // greedy selection
        foodSource[i] = newSol
        fitness[i]=newCost
        unsuccessfulCycles[i] = 0
    end if
end for
```

Fig. 4. Employed bee phase.

C. Onlooker Bee Phase

Onlooker bee calculates the probability *Pi* of choosing a food source using (2). Then sends onlooker bees to a food source(i) according to the associated *Pi* and generates a new food source, as shown in Figure 5.

```
for i = 1 to popSize/2
    newSol = neighborSeach(foodSource[i], i)
    newCost = CalculateFitness(newSol)
        if newCost < fitness[i] // greedy selection
            foodSource[i] = newSol
            fitness[i]=newCost
            unsuccessfulCycles[i] = 0
        end if
end for
```

Fig. 5. Onlooker bee phase.

D. Updating Pheromone Phase

After the onlooker has explored the food source, each bee will deposit the pheromone on the link using (6). Then, the pheromones on all edges are reduced to prevent stagnation. Figure 6 shows the step for deposit and reduction.

```
Deposit pheromone on the links
Reduce pheromone levels on all edges
Memorize the best solution achieved so far
```

Fig. 6. Pheromone update

E. Scout Bee Phase

In the scout bee phase, each bee that couldn't enhance anymore will abandon its food source and search for a new one. Figure 7 shows the scout bee phase algorithm.

```
for i = 1 to popSize/2
    if unsuccessfulCycles [i] > limit
        foodSource[i] = InitializeUsingPheromone()
        newCost = CalculateFitness(newSol)
        fitness[i]=newCost
        unsuccessfulCycles[i] = 0
    end if
end for
```

Fig. 7. Scout bee phase.

F. Neighbor Operator

The neighborhood operator [16-17] is a replacement for (1) to make ABC capable of solving discrete problems. Random swap crossover was used to perform a neighbor search. Figure 8 shows the random swap crossover operation.

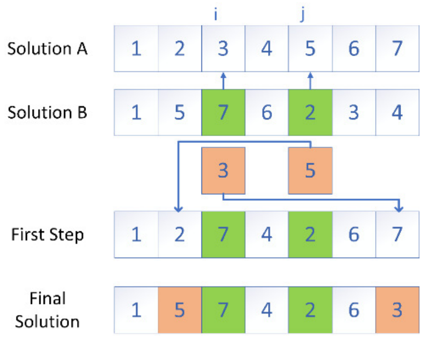

Fig. 8. Random swap crossover.

V. Results and Discussion

The proposed method was implemented using C# and run on a 2.80GHz Intel Core i7 processor with 16GB RAM. The proposed algorithm was tested on 20 TSP instances from [18]. Table I shows the statistical variation of the results for 20 TSPs solved by the two algorithms on 30 independent runs. The results of the experiments demonstrated the superiority of the proposed method over the original ABC, as the error rate and average error rate were better in all cases. Figure 9 shows a comparison between the error rate of ABC and the proposed method, while Figure 10 shows the average rate between the two methods. The error rate was computed using:

$$error\ rate = \frac{best\ value - optimal\ value}{optimal\ value} \times 100 \quad (7)$$

TABLE I. COMPARATIVE EXPERIMENTAL RESULTS ABCUPT WITH ABC

Problem	ABC		Proposed method	
	error	avg error	error	avg error
att48	15.65	25.70	6.15	12.55
eil51	18.54	25.63	6.81	12.49
berlin52	15.87	26.71	6.19	12.18
st70	36.89	53.28	15.06	20.40
eil76	36.62	45.62	12.74	20.97
pr76	39.44	52.80	18.17	27.10
kroA100	74.21	95.10	20.52	31.65
kroB100	65.13	89.32	23.24	35.24
kroC100	79.46	100.86	24.49	36.81
kroD100	72.38	91.55	20.65	31.22
kroE100	63.70	90.23	23.79	36.58
rd100	74.34	86.08	26.99	37.33
eil101	53.58	62.87	24.64	30.86
lin105	82.29	101.68	21.41	33.98
pr107	111.54	172.57	2.40	4.94
gr120	78.68	94.62	34.72	39.89
pr124	143.39	168.81	31.57	37.88
bier127	50.72	70.09	17.25	27.23
ch130	83.01	105.02	20.41	38.49
pr136	99.81	116.42	29.14	42.61

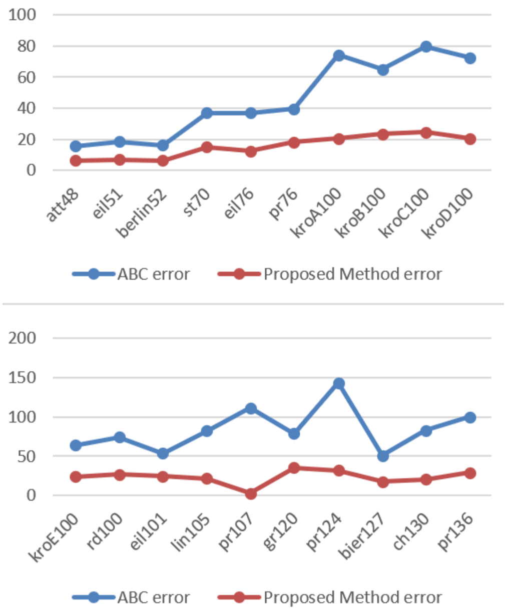

Fig. 9. Comparison between ABC and ABCUPT error rates.

The results show clearly that the proposed method is better than the original ABC. Moreover, using pheromones in ABC gives a memory-like behavior that helps exploration and exploitation but adds more complexity to ABC and increases execution time. In future work, pheromones can be used lightly to improve performance without increasing execution time. Moreover, using a local search algorithm to help bees in exploration could lead to better results.

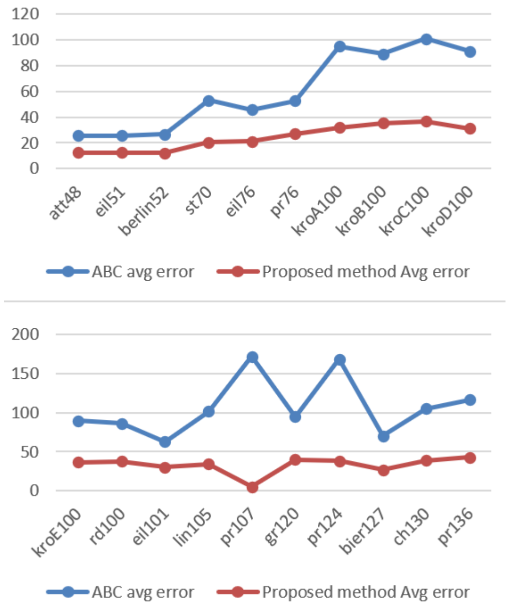

Fig. 10. Comparison of error average rate between ABC and ABCUPT.

VI. Conclusions

This study introduced an enhanced artificial bee colony algorithm using pheromones. The proposed method was tested on 20 TSP problems from TSPLIB. The results were compared with the original ABC and showed that the use of pheromone improved exploration and exploitation but increased execution time. In future work, local search can be integrated to leverage results. Furthermore, the use of pheromones adds memory-like behavior to the ABC algorithm and, therefore, it is easy to integrate the pheromone into other algorithms.

Acknowledgments

The authors acknowledge the financial support of the Universiti Tenaga Nasional under UNITEN-Iraq scholarships and UNITEN BOLD research grant (J510050002/2021150).

References

[1] H. Wei, J. Ji, Y. Qin, Y. Wang, and C. Liu, "A Novel Artificial Bee Colony Algorithm Based on Attraction Pheromone for the Multidimensional Knapsack Problems," in *Artificial Intelligence and Computational Intelligence*, Taiyuan, China, 2011, pp. 1–10, https://doi.org/10.1007/978-3-642-23887-1_1.

[2] J. Ji, H. Wei, C. Liu, and B. Yin, "Artificial Bee Colony Algorithm Merged with Pheromone Communication Mechanism for the 0-1 Multidimensional Knapsack Problem," *Mathematical Problems in Engineering*, vol. 2013, Jul. 2013, Art. no. e676275, https://doi.org/10.1155/2013/676275.

[3] J. M. Moosa, R. Shakur, M. Kaykobad, and M. S. Rahman, "Gene selection for cancer classification with the help of bees," *BMC Medical Genomics*, vol. 9, no. 2, Aug. 2016, Art. no. 47, https://doi.org/10.1186/s12920-016-0204-7.

[4] A. H. Alaidi, C. S. Soong Der, and Y. Weng Leong, "Systematic Review of Enhancement of Artificial Bee Colony Algorithm Using Ant Colony Pheromone," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, Aug. 2021, Art. no. 172, https://doi.org/10.3991/ijim.v15i16.24171.

[5] I. Marouani, A. Boudjemline, T. Guesmi, and H. H. Abdallah, "A Modified Artificial Bee Colony for the Non-Smooth Dynamic Economic/Environmental Dispatch," *Engineering, Technology & Applied Science Research*, vol. 8, no. 5, pp. 3321–3328, Oct. 2018, https://doi.org/10.48084/etasr.2098.

[6] S. D. Chavan and A. V. Kulkarni, "Event Based Clustering Localized Energy Efficient Ant Colony Optimization (EBC_LEE-ACO) for Performance Enhancement of Wireless Sensor Network," *Engineering, Technology & Applied Science Research*, vol. 8, no. 4, pp. 3177–3183, Aug. 2018, https://doi.org/10.48084/etasr.2121.

[7] H. Ehteshami, S. Javadi, and S. M. Shariatmadar, "Improving the Power Quality in Tehran Metro Line-Two Using the Ant Colony Algorithm," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2256–2259, Dec. 2017, https://doi.org/10.48084/etasr.1551.

[8] N. Rahnema and F. S. Gharehchopogh, "An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering," *Multimedia Tools and Applications*, vol. 79, no. 43–44, pp. 32169–32194, Aug. 2020, https://doi.org/10.1007/s11042-020-09639-2.

[9] D. Lei and H. Yang, "Scheduling unrelated parallel machines with preventive maintenance and setup time: Multi-sub-colony artificial bee colony," *Applied Soft Computing*, vol. 125, Aug. 2022, Art. no. 109154, https://doi.org/10.1016/j.asoc.2022.109154.

[10] Z. Han, M. Chen, S. Shao, and Q. Wu, "Improved artificial bee colony algorithm-based path planning of unmanned autonomous helicopter using multi-strategy evolutionary learning," *Aerospace Science and Technology*, vol. 122, Mar. 2022, Art. no. 107374, https://doi.org/10.1016/j.ast.2022.107374.

[11] 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, Nov. 2007, https://doi.org/10.1007/s10898-007-9149-x.

[12] S. ArunKumar, B. V. Kumar, and M. Pandi, "Artificial bee colony optimization based energy-efficient wireless network interface selection for industrial mobile devices," *Computer Communications*, vol. 154, pp. 1–10, Mar. 2020, https://doi.org/10.1016/j.comcom.2020.01.067.

[13] H. Gao, Z. Fu, C.-M. Pun, J. Zhang, and S. Kwong, "An Efficient Artificial Bee Colony Algorithm With an Improved Linkage Identification Method," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4400–4414, Jun. 2022, https://doi.org/10.1109/TCYB.2020.3026716.

[14] C. Twomey, T. Stützle, M. Dorigo, M. Manfrin, and M. Birattari, "An analysis of communication policies for homogeneous multi-colony ACO algorithms," *Information Sciences*, vol. 180, no. 12, pp. 2390–2404, Jun. 2010, https://doi.org/10.1016/j.ins.2010.02.017.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Oct. 1996, https://doi.org/10.1109/3477.484436.

[16] S. S. Choong, L.-P. Wong, and C. P. Lim, "An artificial bee colony algorithm with a Modified Choice Function for the traveling salesman problem," *Swarm and Evolutionary Computation*, vol. 44, pp. 622–635, Feb. 2019, https://doi.org/10.1016/j.swevo.2018.08.004.

[17] M. S. Kıran, H. İşcan, and M. Gündüz, "The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem," *Neural Computing and Applications*, vol. 23, no. 1, pp. 9–21, Jul. 2013, https://doi.org/10.1007/s00521-011-0794-0.

[18] G. Reinelt, "TSPLIB—A Traveling Salesman Problem Library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, Nov. 1991, https://doi.org/10.1287/ijoc.3.4.376.