International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 6, December, 2024, pp. 2161-2167
Journal homepage: http://iieta.org/journals/isi

# Performance Assessment of Three Swarm Intelligence Algorithms in Combinatorial Problem

Khamael Raqim Raheem*, Hafedh Ali Shabat

Department of Computer Networks and Software, Technical Institute of Babylon, Al- Furat Al-Awsat Technical University (ATU), Kufa 51015, Iraq

Corresponding Author Email: khmrakrah@atu.edu.iq

https://doi.org/10.18280/isi.290606

**Received:** 12 August 2024
**Revised:** 30 October 2024
**Accepted:** 4 December 2024
**Available online:** 25 December 2024

***Keywords:***
*combinatorial optimization, meta-heuristic algorithm, Ant algorithm, Bee Colony Optimization, Artificial Bee Colony*

## ABSTRACT

A fast-expanding area of study is the application of metaheuristics to combinatorial problems. The significance of combinatorial optimization problems for both the scientific and industrial domains is the reason behind this. Combinatorial problems are generally challenging due to the many probable varieties that can comprise a reasonable solution. Finding an optimal solution from a determinate set of viable options is called combinatorial optimization. To address combinatorial problems, effective optimization techniques are required. Meta-heuristic optimization employs optimization algorithms to identify optimal solutions for combinatorial problems. This work employs three Meta-heuristic algorithms Ant, Artificial Bee Colony, and Bee Colony Optimization for solving the traveling salesman problem, a famous combinatorial problem. A comparative analysis was carried out between these algorithms after they were examined. The findings of the experiments indicate that the ant method has outperformed other algorithms on TSP in terms of finding the optimal distance (shortest path), also demonstrates that it yielded acceptable performance in terms of computation time.

## 1. INTRODUCTION

The traveling salesman problem (TSP) is extensively researched in operations research and artificial intelligence [1]. The motivation behind the TSP is the problem encountered by a salesperson who needs to visit several customers located in various cities and try for finding the shortest round trip to achieve this mission. The TSP attempts to discover the shortest cyclic path through a directed graph, visiting each node precisely once [2].

Intelligent algorithms own outstanding performance and elevated effectiveness in several fields and the problem nature does not constrain them [3]. Among them, are the cuckoo search algorithm [4], Ant Colony Optimization (ACO) [5], Particle Swarm Optimization (PSO) [6], and Artificial Bee Colony Algorithm (ABC) [7]. These intelligent algorithms it is utilized in different domains like image processing [8], signals processing [9, 10], text mining [11, 12], Water Resources Management [13], Control Systems [14], Transportation Planning [15], combinatorial problems [16], Wireless Sensor Networks [17], VLSI Design [18], and Chemical Process Optimization [19]. Swarm Intelligence refers to the collective behavior of self-organized particles, which is commonly observed in nature. Swarm intelligence approaches utilize a method based on a search process that engages the distribute strategy in which each agent enacts alone [20].

Feasible solutions mean providing all possible outcomes. An optimization problem involves determining the optimal option among all possible outcomes. Swarm intelligence is a probabilistic technique used to solve computational problems and achieve optimal results [21]. Combinatorial problems are difficult to formulate and solve in general, which makes them difficult to tackle. Because there are multiple solvers with various parametrizations, it can also be challenging to select the appropriate "solver algorithm" and determine its optimal configuration. Even for challenging optimization problems, metaheuristics are well known for being effective Combinatorial problem solutions. Because of the vastness of the search space that defines the combinatorial problem in question, they may be the only practical method in some circumstances. The goal of this work is to study the efficacy of optimization algorithms for solving combinatorial problems and analyze the behavior of three famous optimization methods by employing them to solve the TSP.

The remaining content of this paper is systematized as follows: the second section covers the fundamentals of the standard Artificial Bee Colony (ABC) algorithm, Bee Colony Optimization (BCO) algorithm, and Ant Colony Optimization (ACO) algorithm. In the third section, describe the TSP problem. Then, in the fourth section, will data description, and the empirical results and analysis. Lastly, the fifth section produces the conclusion.

## 2. SURROUNDING THEORY

Nature-inspired algorithms have appeared to be viable for resolving exceedingly challenging optimization problems in science and engineering. In recent decades, numerous nature-inspired algorithms have been produced. Nature's chemical, physical, and biological processes are commonly employed to develop nature-inspired algorithms. Furthermore, several algorithms were developed based on sports, sociology, or history [22]. In computer science, numerous algorithms are inspired by natural collective behavior systems. This work will utilize swarm intelligence algorithms, including ACO, ABC, and BCO.

## 2.1 Principle of ACO algorithm

The ACO approach, which Dorigo and Di Caro [23] first presented in the 1990s, is established on the foraging habits of ant colonies. Ants prioritize communal survival over individual species, making them eusocial insects. The communication among them is via touch, sound, and pheromones. The Pheromones are defined as organic chemical molecules released by ants used for a social reaction among individuals of identical species. Ants create pheromone trails on the soil surface, which can be detected by other ants [24]. They live in community nests, and the essential hypothesis of the Ant Colony is to follow the movement of ants from their perches to look for foods in the shortest possible way. starting, ants travel aimlessly to search for food near their nests randomly. The random search creates numerous pathways from the nest to the food sources. Now, based on the quantity and quality of the food, the ants carry an amount of the food back with the required pheromone concentration on its return track. Based on the pheromone trials, the probability of choosing a certain path for the other ants would be a guide aspect of the food source. It appears that both the concentration and the rate of pheromone evaporation determine this probability. Additionally, it can be noticed that the length of each trail can be readily accounted for because the rate of evaporation of pheromone also plays a powerful role [25]. Figure 1 illustrates the behaviors of ants.

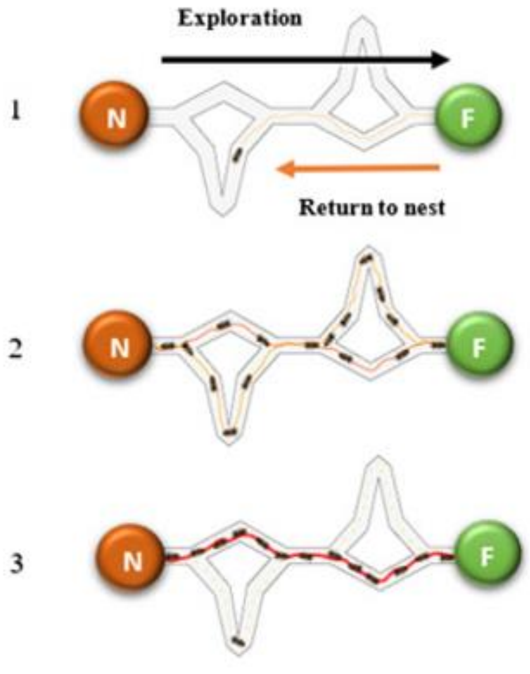

**Figure 1.** Ants choose the shortest path. An ant locates the food sources (F) and then returns to the nest (N) depending on the pheromone spread along its path [24]

The stages of behavior of ants can be described as follows:

Initially, every ant is in the nest. There are no pheromones spread in the environment.

Second stage: Ants will start their search process with an equal probability of each path.

Third stage: Some ants reach sources of food earlier by having a shorter path. Currently, a similar selection dilemma is encountered by the ants, therefore, they will employ the pheromone trail along the shorter route that is already attending, and the likelihood of preferring a certain path is more heightened.

Fourth stage: Most ants return through the shorter track where the concentration of pheromone will enrich. At the same time, the concentration of pheromone along the longer track diminishes due to evaporation, hence the probability of choosing this path will reduce further. Accordingly, the entire colony will utilize the shorter route which has higher probabilities. So, the optimization of the path is achieved.

The steps represent the pseudo code of Ant algorithm [25]:

```
  Procedure Ant algorithm:
    Set up the required parameters and pheromone trials
    while not termination do:
      Generate ant population
      Compute the fitness values relating to every ant
      Locate the best solution by choosing techniques
      Update pheromone trial
    end while
End procedure
```

## 2.2 Principle of ABC algorithm

ABC is a Metaheuristic technique that involves artificial bees collaborating to identify optimal solutions for numerical optimization. This algorithm requires significant resources and a larger computation time [26]. According to Karaboga's 2005 presentation [27], ABC is a type of swarm intelligence that mimics honey bees' food-finding strategies. Honeybees use waggle dances and other techniques to seek out food sources.

The algorithm includes three different bee groups, which are employed bees, onlooker bees, and scout bees, just as they do in nature. Employed bees explore randomly and return to the hive with information about the surroundings. This exploratory search information is shared with onlooker bees. The onlooker bees use a probabilistic method to begin a neighborhood search. Meanwhile, the scout bees do a random search before carrying out the exploitation [28]. Bees navigate a multidimensional search space, selecting food sources based on their expertise and partners before determining their location. Randomly, Bees select a source of food without previous experience or knowledge. If a new source provides more nectar than the previous source, the bee will remember the new location and replace the old one. This algorithm combines local search approaches from employed and onlooker bees with a global search process executed by Scouts and onlookers [29]. Inside a hive, onlookers and employed bees will be split equally. Where the number of employed bees is equivalent to the available food sources. Each food source has a single employed bee. The quality of food sources in a given place correlates with fitness value [30]. The strategy used via bees to locate food source is applied to determine the optimal solution [31]. The initial stage involves randomly assigning food source locations. Eq. (1) indicates this stage [16].

$$F(x_i), x_i \in R^D, i \in \{1,2,\dots,S_N\} \quad (1)$$

where, $x_i$ is a D-dimensional vector which represents the position of food sources, ($F(x_i)$) symbolizes the objective function, ($F(x$i)) symbolizes the objective function, which defines the optimal solution, and the number of the food source ($S_N$) is given. After that, the population undergoes three key steps: first, update possible solutions; second, pick feasible solutions; in the third step, avoid suboptimal solutions. Every employed bee determines a new candidate food source place for updating potential solutions. The neighbors of the food source that was previously chosen are the basis for the decision. To determine the location of the new food source, utilize Eq. (2).

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \quad (2)$$

where $v_{ij}$ refers to a new feasible solution which updated. $x_{ij}$ represents the previous solution, $x_{kj}$ refers to the neighboring solution, and $\emptyset_{ij}$ is a random number between -1 and 1. $K \in \{1,2,\dots,SN\}$ and $j \in \{1,2,\dots,D\}$. The notation ($x_{ij}$-$x_{kj}$) indicates a difference in location along a given dimension. Depending on the fitness value, if the new position is higher than that of the previous one, the location of the old food source, which was stored in the employed bee's memory, will be replaced by the new location. When returning to the hive, employed bees will then share with the onlooker bee the fitness value of new food sources. Each onlooker bee chooses one of the suggested food sources based on the fitness value acquired from the employed bee. The likelihood of selecting a food source is shown by Eq. (3).

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (3)$$

where, $P_i$ represents the probability of an onlooker bee selecting of source of food [32]. $f_{iti}$ describes the fitness value of the food source(i). Afterward, the Onlooker type visits the determined source of food and select a new candidate position in the surrounding area. In the third step in order to avoid suboptimal solutions, food sources that do not improve the fitness value will be departed and modified into a new position appointed randomly by the scout bees [26]. Eq. (4) used to compute the new location of the Scout Bee [16].

$$x_{ij} = x_j^{min} + rand[0,1] * (x_j^{max} - x_j^{min}) \quad (4)$$

A lower bound of the location of the food source is indicated by $x^{min}$, whereas an upper bound of the location of the food source is indicated by $x^{max}$, within the dimension $j$ [33]. The stop condition is represented by the Max cycle number (MCN) where utilized to restrict the iterations count. The search procedure will be continued until the outcome of the objective function satisfies a predetermined threshold or the iteration count reaches the MCN. The threshold value is elected to be identical to either the global minimum or maximum value according to the natural optimization problem being handled [34].

### 2.3 Principle of BCO algorithm

BCO is an algorithm that is based on population. It more closely resembles natural bee behavior. This algorithm differs from others by utilizing scout bees, emphasizing the importance of hive location, and adopting a more natural recruitment strategy. BCO involves artificial bees cooperating to tackle complex combinatorial optimization problems [35].

The search operation begins with all artificial bees present in the hive; they now communicate with each other throughout the search operation. Every bee carries out a succession of local moves and constructs a solution for the problem gradually. Until one or more feasible solutions are constructed, Bee continues to add solution components to the current partial solution. The search process of the BCO approach runs through multiple iterations. When bees construct one or more possible solutions, the first iteration is considered to be over. After that, saving the best solution met at this iteration, and then the second iteration will begin. Depending on this strategy, Bees develop solutions gradually during the second iteration and so on and continue in the succession of iterations. lastly, Each iteration yields a partial solution [36].

The search inside the BCO algorithm is based on a cooperation strategy between bees to locate the optimal solution. Where the bees fly within search space and each one generates one solution during forward pass or backward pass. The bees by the forward pass explore the search space and create a diverse partial solution where it uses several moves determined previously for creating or enhancing the solution, constructing a new partial solution. The bees perform this through a combination of personal exploration and collaborative previous knowledge.

Bees will return to the hive meanwhile the backward stage and exchange information about the solutions inside the hive among entire bees. This means each bee will participate in making the decision. An objective function will compute, and the possible solutions evaluated. Depending on particular probability the bee will decide to reject the partial solution or not, and evolve into an uncommitted follower, persist in expanding the recently formed partial solution without conscripting the nests, or dance for recruiting the nest before backing to the recently created partial solution. According to solutions quality, each bee reveals a specific degree of adherence to the pathway that guides to the formerly discovered partial solution. The quality of the created partial solutions determines this degree of loyalty. The bees will utilize a second forward pass to extend formerly created partial solutions, and after that, again replicate the backward pass correspondingly before returning to the hive. Inside the hive, bees will participate in decisions again. Then running a third forward pass, and so on. Eventually, the iterations will stop when one or more feasible solutions are located [37].

The algorithm uses forward and backward pass stages alternating to generate all feasible solutions [38]. After all solutions are accomplished, the algorithm picks the best one and employs it for modifying the global solution. The rest of the solutions are ignored currently. Then, initiate a new iteration. This procedure will repeated till an ending condition is met( the time of CPU which is permitted, or the max count of forward-backward passes, or any other condition). Ultimately, the global solution (optimal solution) which is obtained is stated as the final solution.

Combinatorial optimization problems are addressed by the BCO technique in steps, with one optimization variable standing employed per step [39]. Assume to describe the number of pre-steps as follows ST=($st_1$ , $st_2$ ,..., $st_m$ ), where m signifies the number of steps, also, suppose B represents the bee's count functioned in the search process, I refers to the total number of iterations, and $S_j$ ($j$=1,2,...,$m$) denotes a gathering of partial solutions for stage $st_j$. Figure 2 shows the passes of the BCO algorithm [35].

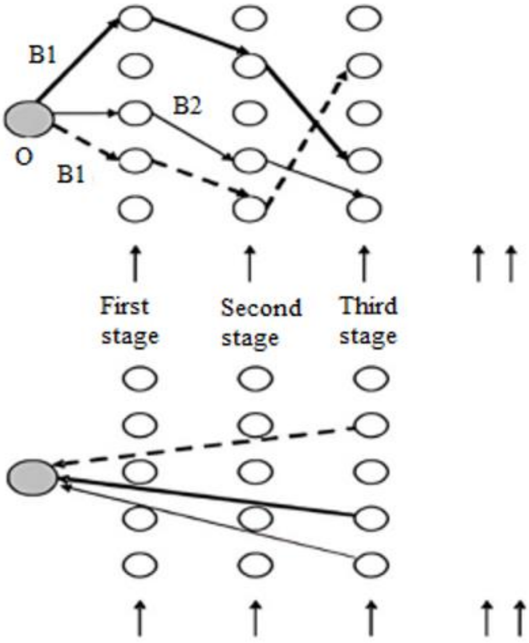

**Figure 2.** Forward passes and backward passes [35]

The following steps describes the passes of BCO algorithm [35].

Step one: start with the numbers of bee (B) and the number of iterations I, choose the set of stage ST=($st_1$,$st_2$,….,$st_m$), which signifies the productive movements number through the one forward pass.
Then, determine any feasible solution x of the problem (this solution denotes the preliminary solution)
Step two: initiating i:=1, Implement the successive steps until i=I.
Step three: initiating j:=1, Repeat the subsequent steps until j=m, (m is the number of stages)
Forward pass: Permit bees depart the hive and generate partial solutions (B) from Sj at stage $st_j$.
Backward pass: All bees go back to the hive. After that Intercommunicate with one another about the partial solution quality then becomes an uncommitted follower.
Extend the same partial solution without attracting nestmates, or dance to recruit nestmates, before returning to the generated partial solution.
Step four: if a new best solution $x_i$ is better than the current best solution x, should be update the current best solution ($x:=x_i$)
Step five: Increases the I, i=i+1.

## 3. DATA SETUP

The traveling salesman issue involves determining the most efficient path that visits each city in a list of cities precisely once, while also returning to the starting city. This is done by considering the distances between each pair of cities. In this study, we suppose have n = 25 cities generated randomly, each city represented by one point has coordinates in an Euclidean space, as shown in Figures 3-4 which represent the distribution of cities in 2D and 3D space, respectively. In actuality, the permutations of the routing order of n cities in the TSP problem are computed as $(n-1)!/2$, which leads to impossible computing the permutations in a suitable of time. particularly, if the value of n is great. Consequently, we utilized three optimization techniques to get the optimal solution, which is the shortest path with the least amount of time. The purpose of using the TSP problem in this study is solely to demonstrate the characteristics of the executed optimization algorithms by comparing their performance.

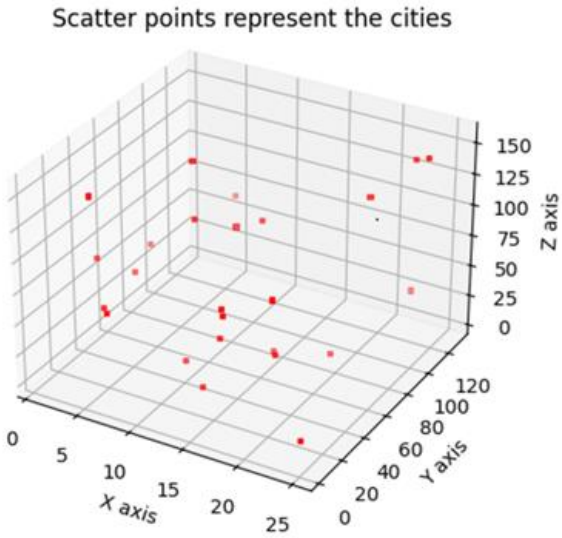

**Figure 3.** Distribution of data in 3-dimensional space

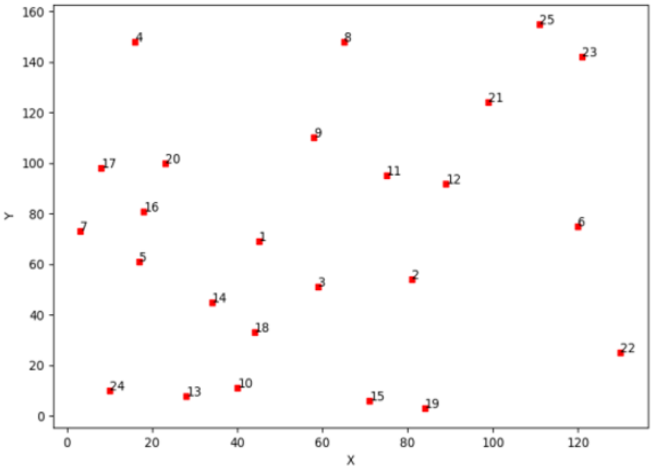

**Figure 4.** Distribution of 2-dimensional data

In actuality, the permutations of the routing order of n cities in the TSP problem are computed as $(n-1)!/2$, which leads to impossible computing the permutations in a suitable of time. particularly, if the value of n is great. Consequently, we utilized three optimization techniques to get the optimal solution, which is the shortest path with the least amount of time. The purpose of using the TSP problem in this study is solely to demonstrate the characteristics of the executed optimization algorithms by comparing their performance.

## 4. EXPERIMENTAL AND RESULTS

In this work, three experiments were conducted to show the effectiveness of ACO, BCO, and ABC algorithms by comparing the performance of the capability to solve the TSP. These methods were executed employing a Python program by JetBrains PyCharm Edition 2023.3 and running by a machine that has a CPU (Core i5) with 2.5 GHz, and memory of 12 GB. The data was initially generated randomly and included numerous points, each of which symbolized a city. A total of 25 cities were determined. The data allocation is illustrated in Figures above.

### 4.1 First experiment

A group of k ants is constructed to make an "ant colony," and a randomly chosen city as the beginning point (The initial town is chosen without any specific reason as a whole journey might commence from any city). The ACO Algorithm has parameters that should be fitted, to achieve adequate performance. where α defines the weights ascribed to pheromone concentration, β refers to ant visibility, ρ implies the evaporation rate of the pheromone, which is utilized to modify the pheromone, and Q denotes a pheromone amplification constant. In this experiment, parameters were setup as follows: number of ants (k=10), number of iterations =120, α=1, β =1, ρ=0.5, and Q=1. The algorithm located the shortest route via travel of the following cities [15, 19, 22, 6, 2, 3, 1, 9, 11, 12, 21, 23, 25, 8, 4, 17, 20, 16, 7, 5, 14, 18, 10, 13, 24] with the distance= 691.6827218570513 and Elapsed time is 18.797309 seconds. Figure 5 displays a diagram of the path that is discovered by the ACO algorithm.

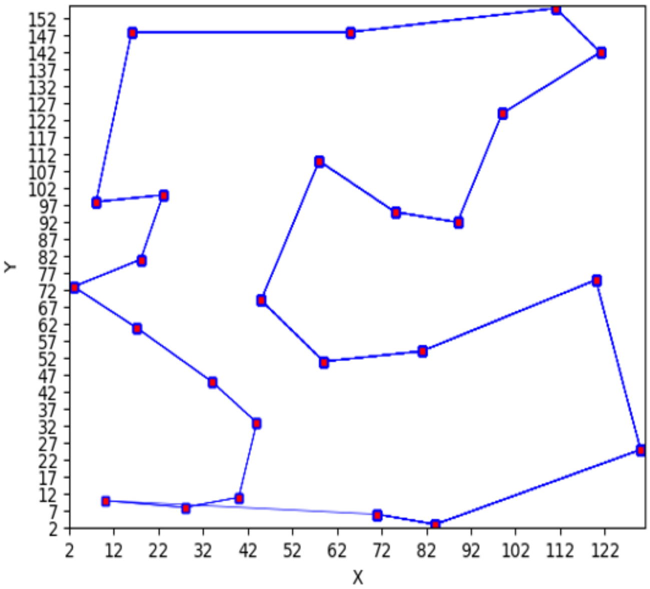

**Figure 5.** An illustration of the best route utilizing ACO algorithm

### 4.2 Second experiment

The experiment aims to evaluate how well BCO solved the TSP issue. The algorithm requires control parameters to act and perform appropriately. The first parameter value is the maximum number of cycles (MCN), which is taken in this experiment to be 1500. The second value is the maximum size of the population which set is at 120. The count of runs of parameters was appointed to 80, where there is MCN for each run. Also, there is another parameter that is Dimension which depends on the count of cities. In this experiment, the bees were split as follows: the employed bees with 50 percent, onlooker bees with 50 percent, and 20 percent scout bees. The ABC located the following route by travels through the cities [3, 5, 2, 18, 14, 9, 12, 23, 13, 4, 6, 15, 16, 24, 22, 8, 19, 1, 21, 17, 0, 10, 11, 20, 7] with distance = 1115.597 and elapsed time =1327.359542 seconds as illustrated in Figure 6.

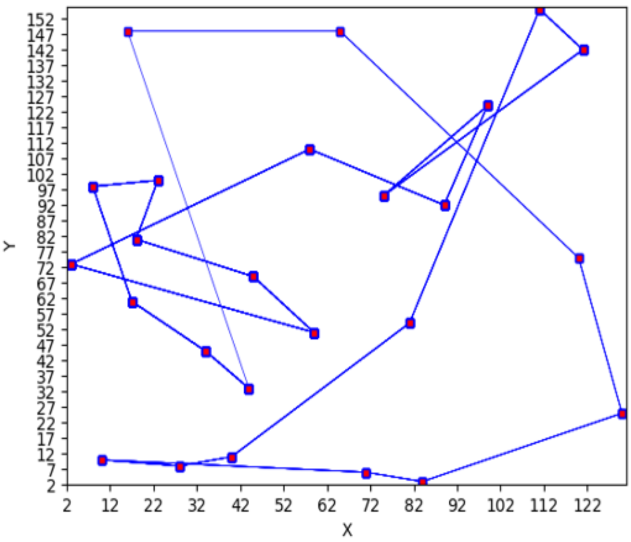

**Figure 6.** An illustration of the optimal route utilizing ABC algorithm

### 4.3 Third experiment

This experiment was executed to assess the effectiveness of BCO in solving the TSP matter. The settings of parameters are specified as follows: the count of bees is set to 300, a number of stages is set to 3, and the percentage of bee probability to be scout type is 10%, after conducting an experiment the BCO found that the path is through travel the cities [18, 14, 5, 17, 20, 16, 1, 3, 7, 9, 12, 21, 11, 23, 25, 2, 10, 13, 24, 15, 19, 22, 6, 8, 4] with distance =991.9184714895329 and Elapsed time= 31.026371 seconds as displayed in Figure 7.

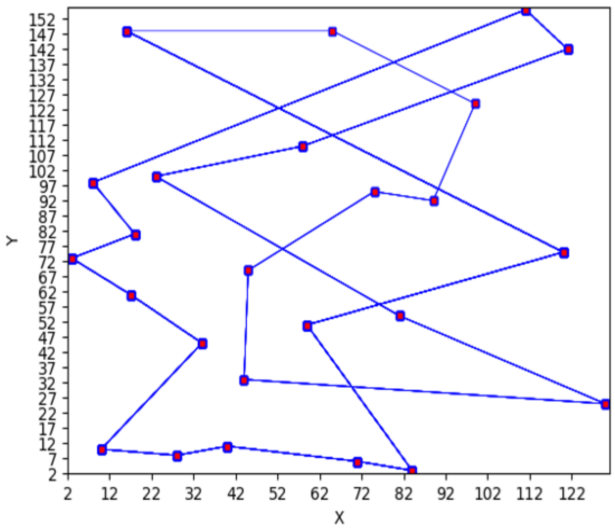

**Figure 7.** An illustration of the optimal route utilizing the BCO algorithm

In the ACO algorithm, ants start by choosing cities and walk randomly through them. Each Ant constructs a solution depending on restriction where each town has to be visited precisely once. According to experiments ACO has given the best result where produced the shortest path with less computational time. Due to the truth, the next city is chosen based on the pheromones associated with each edge between two points, and a few other parameters. therefore, it permits dynamic rerouting via the shortest track if one node is broken. While the other algorithms instead suppose that there are the network is static. Therefore, It gives positive feedback which leads to the discovery of good solutions within a reasonable computing time. where yielded the shortest path with slight time compared with the ABC and BCO algorithms. The ABC uses a greedy selection process that combines a local and a global search strategy to obtain the optimal solution. Hence, it needs a lot of resources and large computation time. In contrast, the BCO technique is based on collaboration, where Bees can be more effective when working concurrently. BCO yielded good results and it needed reasonable computing time to identify the optimal solution compared with ABC. lastly, must be mentioned, that due to supplementary limitations like time, cost, and client preference, real-world applications call for transformations to satisfy their achieving.

## 5. CONCLUSIONS

Combinatorial problems are challenging due to the many probable varieties that can comprise a reasonable solution. The best strategy to tackle a combinatorial problem is considering all possible solutions in the search area. Yet, checking all of the viable solutions is not always practicable, particularly when the search area is extensive. Many meta-heuristic methods have been developed and adapted to address these problems. Meta-heuristic techniques aren't guaranteed to locate the optimal solution, but they do strive to explore diverse sections of the search area in a clever way to obtain a near-optimal solution at a lower cost time. In this work, the Metaheuristic algorithms ACO, ABC, and BCO enlightened from natural systems were employed for solving the TSP combinatorial optimization problem. It has been demonstrated that all employed methods perform effectively on the TSP problem. ACO is regarded as moderately easy-to-implementation due to well-defined updating rules of pheromone, and easy application. Where ACO needed to adjust a few parameters such as pheromone concentration and evaporation rate, and exploration-exploitation balance. Therefore, produced a good result with less computational time. whereas, ABC and BCO require more complicated strategies to mimic the behavior of bees accurately. Both, require more parameters related to the search behavior of bees and communication among them. Where ABC algorithm uses the greedy selection technique for acquiring the optimal solution, accordingly, this algorithm requests excessive processing resources and also excessive computational time. In contrast, BCO approach is based on collaboration where artificial bees can be more effective when they operate together. Consequently, they could give a reasonable solution in an appropriate computing time compared with the ABC algorithm. Finally, the experiments revealed that metaheuristic algorithms are appropriate to tackle combinatorial problems such as the TSP problem, and can be applied to solve other problems in future works.

## REFERENCES

[1] Yang, X.S., Karamanoglu, M. (2020). Nature-inspired computation and swarm intelligence: A state-of-the-art overview. Nature-Inspired Computation and Swarm Intelligence, pp. 3-18. https://doi.org/10.1016/b978-0-12-819714-1.00010-5

[2] Voss, S. (2006). Book review: Holger H. Hoos and Thomas Stützle: Stochastic local search: Foundations and applications (2005). Mathematical Methods of Operations Research, 63(1): 193. https://doi.org/10.1007/s00186-005-0051-3

[3] Li, X., Zhang, S., Shao, P. (2024). Discrete artificial bee colony algorithm with fixed neighborhood search for traveling salesman problem. Engineering Applications of Artificial Intelligence, 131: 107816. https://doi.org/10.1016/j.engappai.2023.107816

[4] Yang, X.S., Deb, S. (2010). Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation, 1(4): 330-343. https://doi.org/10.1504/ijmmno.2010.035430

[5] Shtovba, S.D. (2005). Ant algorithms: Theory and applications. Programming and Computer Software, 31: 167-178. https://doi.org/10.1007/s11086-005-0029-1

[6] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, pp. 1942-1948. https://doi.org/10.1109/icnn.1995.488968

[7] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 200: 1-10.

[8] Oliva, D., Abd Elaziz, M., Hinojosa, S. (2019). Multilevel thresholding for image segmentation based on metaheuristic algorithms. Metaheuristic Algorithms for Image Segmentation: Theory and Applications, 59-69. https://doi.org/10.1007/978-3-030-12931-6_6

[9] Salman, H.M., Al-Qurabat, A.K.M. (2021). Bigradient neural network-based quantum particle swarm optimization for blind source separation. IAES International Journal of Artificial Intelligence, 10(2): 355. https://doi.org/10.11591/ijai.v10.i2.pp355-364

[10] Salman, H.M., Abbas, N.A. (2021). Comparative study of QPSO and other methods in Blind Source Separation. Journal of Physics: Conference Series, 1804(1): 012097. https://doi.org/10.1088/1742-6596/1804/1/012097

[11] Shabat, H.A., Abbas, N.A. (2020). Enhance the performance of independent component analysis for text classification by using particle swarm optimization. In 2020 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, pp. 1-6. https://doi.org/10.1109/icoase51841.2020.9436547

[12] Ibrahim, M.A., Al-Tahar, I.A., Salamah, H.M., Mohamad, N.I. (2024). Improving quality of service in cloud computing frameworks using whale optimization algorithm. Ingénierie Des Systèmes d Information, 29(5): 1949-1957. https://doi.org/10.18280/isi.290526

[13] Niu, W., Feng, Z., Jiang, Z., Wang, S., Liu, S., Guo, W., Song, Z. (2021). Enhanced harmony search algorithm for sustainable ecological operation of cascade hydropower reservoirs in river ecosystem. Environmental Research Letters, 16(5): 055013. https://doi.org/10.1088/1748-9326/abf60c

[14] Dogruer, T. (2022). Grey wolf optimizer-based optimal controller tuning method for unstable cascade processes with time delay. Symmetry, 15(1): 54. https://doi.org/10.3390/sym15010054

[15] Saju Sankar, S., Vinod Chandra, S.S. (2020). A multi-agent ant colony optimization algorithm for effective vehicular traffic management. In Advances in Swarm Intelligence: 11th International Conference, ICSI 2020, Belgrade, Serbia, pp. 640-647. https://doi.org/10.1007/978-3-030-53956-6_59

[16] Shabat, H.A., Raheem, K.R. (2023). Analysis study of the bee algorithms as a mechanism for solving combinatorial problems. Indonesian Journal of Electrical Engineering and Computer Science, 30(2): 1091-1098. https://doi.org/10.11591/ijeecs.v30.i2.pp1091-1098

[17] Harikrishnan, R., Jawahar Senthil Kumar, V., Sridevi Ponmalar, P. (2016). Firefly algorithm approach for localization in wireless sensor networks. In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, pp. 209-214. https://doi.org/10.1007/978-81-322-2529-4_21

[18] Zhu, L.C., Yang, R.P., Chen, M.X., Jia, X.M., Li, X.X., Du, S.M. (2012). An efficient simulated annealing based VLSI floorplanning algorithm for slicing structure. In 2012 International Conference on Computer Science and Service System, Nanjing, China, pp. 326-330. https://doi.org/10.1109/csss.2012.89

[19] Fan, Q., Wang, W., Yan, X. (2017). Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems. Applied Soft Computing, 59: 33-44. https://doi.org/10.1016/j.asoc.2017.05.044

[20] Yang, L., Qi, J., Xiao, J., Yong, X. (2014). A literature review of UAV 3D path planning. In Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp. 2376-2381. https://doi.org/10.1109/wcica.2014.7053093

[21] Deepa, O., Senthilkumar, A. (2016). Swarm intelligence from natural to artificial systems: Ant colony optimization. Networks (Graph-Hoc), 8(1): 9-17. https://doi.org/10.5121/jgraphoc.2016.8102

[22] Osaba, E., Diaz, F., Onieva, E. (2013). A novel meta-heuristic based on soccer concepts to solve routing problems. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, New York, United States, pp. 1743-1744. https://doi.org/10.1145/2464576.2480776

[23] Dorigo, M., Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406): Washington, DC, USA, pp. 1470-1477. https://doi.org/10.1109/cec.1999.782657

[24] Hemmati-Sarapardeh, A., Larestani, A., Menad, N.A., Hajirezaie, S. (2020). Applications of Artificial Intelligence Techniques in the Petroleum Industry. Gulf Professional Publishing.

[25] Rezvanian, A., Vahidipour, S.M., Sadollah, A. (2023). An overview of ant colony optimization algorithms for dynamic optimization problems. Optimization Algorithms-Classics and Recent Advances. https://doi.org/10.5772/intechopen.111839

[26] Huo, J., Liu, L. (2018). An optimization framework of multiobjective artificial bee colony algorithm based on the MOEA framework. Computational Intelligence and Neuroscience, 2018: 5865168. https://doi.org/10.1155/2018/5865168

[27] Yuce, B., Packianather, M.S., Mastrocinque, E., Pham, D.T., Lambiase, A. (2013). Honey bees inspired optimization method: the bees algorithm. Insects, 4(4): 646-662. https://doi.org/10.3390/insects4040646

[28] Junaedi, D., Maulidevi, N.U. (2011). Solving curriculum-based course timetabling problem with artificial bee colony algorithm. In 2011 First International Conference on Informatics and Computational Intelligence, Bandung, Indonesia, pp. 112-117. https://doi.org/10.1109/ici.2011.28

[29] Boopathi, M., Sujatha, R., Senthil Kumar, C., Narasimman, S. (2017). Quantification of software code coverage using artificial bee colony optimization based on Markov approach. Arabian Journal for Science and Engineering, 42: 3503-3519. https://doi.org/10.1007/s13369-017-2554-7

[30] Hsu, W.Y., Hu, Y.P. (2015). Artificial bee colony algorithm for single-trial electroencephalogram analysis. Clinical EEG and Neuroscience, 46(2): 119-125. https://doi.org/10.1177/1550059414538808

[31] Celik, Y. (2021). An enhanced artificial bee colony algorithm based on fitness weighted search strategy. Automatika, 62(3–4): 300-310. https://doi.org/10.1080/00051144.2021.1938477

[32] Banharnsakun, A. (2017). A MapReduce-based artificial bee colony for large-scale data clustering. Pattern Recognition Letters, 93: 78-84. https://doi.org/10.1016/j.patrec.2016.07.027

[33] Banharnsakun, A., Achalakul, T., Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing, 11(2): 2888-2901. https://doi.org/10.1016/j.asoc.2010.11.025

[34] Teodorovic, D., Dell’Orco, M. (2005). Bee colony optimization–A cooperative learning approach to complex transportation problems. Advanced OR and AI Methods in Transportation, 2005: 51-60.

[35] Teodorovic, D., Lucic, P., Markovic, G., Dell'Orco, M. (2006). Bee colony optimization: principles and applications. In 2006 8th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, pp. 151-156. https://doi.org/10.1109/neurel.2006.341200

[36] Forsati, R., Keikha, A., Shamsfard, M. (2015). An improved bee colony optimization algorithm with an application to document clustering. Neurocomputing, 159: 9-26. https://doi.org/10.1016/j.neucom.2015.02.048

[37] Todorovic, N., Petrovic, S. (2012). Bee colony optimization algorithm for nurse rostering. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(2): 467-473. https://doi.org/10.1109/tsmca.2012.2210404

[38] Girsang, A.S., Tsai, C. W., Yang, C.S. (2012). A fast bee colony optimization for traveling salesman problem. In 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications, Kaohsiung, Taiwan, pp. 7-12. https://doi.org/10.1109/ibica.2012.44

[39] Dell'Orco, M., Marinelli, M., Altieri, M.G. (2017). Solving the gate assignment problem through the fuzzy bee colony optimization. Transportation Research Part C: Emerging Technologies, 80: 424-438. https://doi.org/10.1016/j.trc.2017.03.019