Babylonian Journal of Networking
Vol.2025, **pp**. 134–142
DOI: https://doi.org/10.58496/BJN/2025/012; ISSN: 3006-5372
https://mesopotamian.press/journals/index.php/BJN

Research Article

# ACO-Based Path Optimization Inspired by TSP for Routing Efficiency in Communication Networks

Mohammed Amin[1,*], Mohamed Chahine Ghanem [2]

[1] *King Abdullah the II IT School, The University of Jordan, Amman 11942, Jordan*

[2] *Cybersecurity Institute, School of Computer Science and Informatics, University of Liverpool, L69 3BX, Liverpool, UK*

**ARTICLE INFO**

**Article History**

Received 17 Aug 2025
Revised 15 Sep 2025
Accepted 15 Oct 2025
Published 25 Oct 2025

**Keywords**
ACO,
TSP,
Routing Optimization,
Communication Networks,
Candidate Set,
Path Selection,
Adaptive Parameters,
Network Congestion,
Heuristic Algorithms.



**ABSTRACT**

Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem, and it is generally concerned with searching for this shortest tour that passes through each of a collection of cities and returns to the point of departure. In recent years the properties of TSP have proved to be very transferable to network routing problems where the ability to select optimal paths is imperative for maintaining low congestion and efficient communication. In this paper, we have studied the application of the established swarm-based optimization algorithm, Ant Colony Optimization (ACO), for solving routing problems in communication networks under a basic TSP-inspired model. The research covers the critical issues such as network congestion, routing delay and slow convergence scenario that are often present in dynamic network scenario. In order to improve the performance of the ACO, the proposed algorithm combines candidate set to assess path quality and applies the mechanism of adaptive adjustment of parameters to enhance the search and accelerate the convergence. The approach is suitable for routing in various simulated network scenarios and is shown to be more efficient and stable.

## 1. INTRODUCTION

The TSP is one of the most known and studied combinatorial optimization problems which consists on finding a shortest path to visit a set of cities once and then go back to the initial city. Although the problem seems simple to understand, its computational complexity is exponential in the number of nodes and thus it is belonging to NP-hard class [1]. Apart from its theoretical mystique, TSP has found practical applications in areas as diverse as logistics and delivery routing, circuit board fabrication, and, of particular relevance here: network routing [2].In the application-oriented wireless network like WSN, MANET, and IoT infrastructure, the network topology changes frequently, and how to optimize routing becomes a thorny issue. Efficient routing algorithms are expected to reduce the latency, avoid bottlenecks, and be tolerant to network changes, which is common with TSP [3]. This analogy has motivated the introduction of TSP-inspired models also in routing algorithms to improve path selection, resource consumption and QoS [4]. Several metaheuristic algorithms have been suggested to solve TSP as well as its network-based counterparts. They naturally motivated new approaches for problem solving, among which were the Ant Colony Optimization (ACO) algorithm developed by Dorigo et al. [5], is remarkable for its bio-inspired properties. ACO models the act of the pheromone laying and the trail following performed by ants while searching for food. In the approach, virtual ants search for potential structures by laying virtual pheromones, to let the swarm suitor towards optimal or near optimal ways. This emergent behavior of an assembly is used by ACO in solving TSP successfully [6]. Standard ACO algorithms have some limitations although it is a successful method in large-scale or dynamic environments. Typically encountered problems are premature convergence, slow discovery of optimal solutions, and local minima often being difficult to escape [7]. These constraints become critical in networking environment where dynamic adaptiveness in real-time and routing efficiency are a must [8]. In this paper, ACO will be used to improve ACO for TSP-motivated routing optimization over communication network. More specifically, we address the Embedding of the Candidate Set methodology to enhance the solution and to speed up convergence [9]. Weary also introduce dynamic parameter

*Corresponding author. Email: mohamed.chahine.ghanem@liverpool.ac.uk

adaptation techniques based on entropy to balance exploration and exploitation during the searchρ"106b8 process [10]. We compare our method to both traditional ACO and other types of heuristics over a range of network-based routing scenarios. The performance (in convergence time, quality of solution, and routing efficiency) results indicate that the proposed model is a promising model in the area of intelligent network optimization [11]. Ultimately, this work contributes to join the theoretical combinatorial optimization and its ap- plication problems in networking, and providing better solutions for routing protocols which are essential for current communication systems, logistics, and infrastructures [12].

## 2. L ITERATURE REVIEW

The TSP has been a standard test problem for combinatorial optimization algorithms for a considerable period. The ease of its expression belies its computational complexity; several suggestions have been published to seek optimal or near-optimal solutions [13-17]. One of the well-known approaches in using the above characteristics is Ant Colony Optimization (ACO) which inspired by how real ants search for food, exploit available resources and how they send signals to communicate with other ants to optimize paths [18]. The essential pioneering work on ACO performed by [19] was the invention of the Ant System (AS). In these methods, a set of simulated ants imitate the behavior of the real ants, pheromones are emitted by ants to drive the future solution building. The quantity of pheromone on any given path affects the likelihood of being adopted by the subsequent ants, and facilitating over time a collective learning process. The Ant System has also been shown to have better results than many original optimization algorithms, mainly on the TSP, both in convergence speed and quality of the obtained solution. In later works, authors tried to optimize ACO by reducing computational burden and increasing convergence stability. The Candidate Set method was first introduced by [20] aiming to directly limit the set of edges the ants may choose to add. In contrast with exact solution generation methods, where potentially useful paths are treated equally, this more-narrow search over a subset of highly promising paths reduces the search space, reduces computation time, and generally yields higher-quality solution [21-24]. The Candidate Set approach has been particularly useful in the context of large-scale TSP instances for which full exploration is not feasible. In addition to structural improvements, adaptive methods have been incorporated to increase the dynamism of ACO. Among them, the entropy-based parameter adaptation is an attractive approach. The optimization of algorithm parameters adaptively to the diversity of solutions expressed in the form of entropy in this work allows the algorithm to orchestrate exploration and exploitation. This eases premature convergence and increases the likelihood of better solutions [25-27]. In this regard, showed that adaptive parameter control could drastically increase the robustness and the efficiency of ACO to solve TSP. A significant improvement in ACO technique was due to the invention of the Ant Colony System (ACS) by Dorigo and Gambardella. Contrary to the basic Ant System, ACS utilizes local search and global pheromone updates. The local search step of the method cleans the partial solutions and makes small profound modifications (e.g., switching nodes and changing the path segments) in order to improve the whole path [28-31]. These improvements render ACS particularly suitable for complex routing problems as those arising in communication networks. In general, the literature shows a development from the basic Ant System toward more advanced methods such as ACS, the Candidate Set method, and the entropy-based dynamic adaptation. Collectively, such advances have translated into better solution quality, faster convergence and strong algorithmic performance for TSP and its network-inspired analogues [29,30]. The correspondence of TSP and path optimization in dynamic network environment makes such developments very attractive for ACO applications to routing optimization in communication networks [31-36].

## 3. PROPOSED ENHANCEMENTS

In order to enhance the efficiency and flexibility of Ant Colony Optimization (ACO) algorithms for the Traveling Salesman Problem (TSP), and thus breaking into communication network optimization, two novel developments are proposed in the current work, based on the latest literature progress:

### 3.1 Candidate Set-Based Construction Strategy

The first modification is the application of a Candidate Set technique which confines the ants to use in each iteration a restricted and therefore focused subset of high potential edges in their solution building process. This technique is designed to:

a) Decrease computational cost by limiting the number of edges that each ant has to assess, and hence speed up the solution construction.

b) Concentrate the search on more positive route, as Candidate Sets are usually pre-processed according to heuristics such as geographical distance, or historical concentration of pheromone.

c) Integrate a hybrid selection mechanism where ants do not only consider pheromone concentration and heuristic information to probabilistically select the next node, but restrict the search for the next node to those in the candidate list.

The algorithm is advantageous for the following reasons: 1) the ants are guided through a narrow part of the solution space, which leads to faster convergence and makes the algorithm less susceptible of being trapped into poorer global solutions and 2) the heuristic information (such as distances as paths are taken) is not corrupted by the history of ants.

### 3.2 Entropy-Based Adaptive Parameter Tuning

The second proposed improvement is a dynamic adaptation of the algorithm parameters during runtime based on the entropy. Entropy is used to evaluate uncertainty or diversity of pheromone distribution and solution space. The approach involves:

a) Assessing the entropy levels during optimization to ensure no premature convergence or stagnation.

b) Dynamically tuning alpha, beta and pheromone evaporation parameter based on entropy variations, trying to balance the explorative and exploitative tendencies.

c) Promoting the exploration of new areas in the solution space when the algorithm starts to converge on suboptimal path(s), thereby enhancing the robustness and global optimality of the final solution.

This adaptive strategy helps the ACO algorithm to prevent premature convergence and adapt to more intricate, high-dimensional optimization landscapes, which frequently appear in dynamic routing scenarios.

The combination of entropy-based parameter adaptation with the Candidate Set strategy is intended to improve the efficiency, adaptability, and convergence properties of ACO to TSP-like routing problems in communication networks. We show the improvements of these enhancements in our experiments in the following sections of this paper.

## 4. METHODOLOGY

This study used Ant Colony Optimization (ACO) to solve the TSP to enhance the routing efficiency for the communication network. The proposed approach incorporates two important modifications: the Candidate Set method for computation load reduction and an entropy-based dynamic parameter learning technique to prevent early convergence and improve global search capability. Our method consisting of several stages including raw data collection, preprocessing and parameter initialization, solution searching and performance assessment.

### 4.1 Data Collection and Preprocessing:

Finally, for evaluating the robustness and applicability of the proposed method, the TSPLIB [31], which is commonly used to obtain benchmark instances of the TSP, was applied. From small to large-scale problems, the size of these instances range between 10 to 50 to 1000 cities. The tested datasets involve both symmetric and asymmetric instances to represent different routing conditions. Each occurrence is a list of the cities where each city is a spot defined by its x and y. Where d (i, j) is the Euclidean distance between cities i and j:

$$d_{ij}=\sqrt{(xi-xj)2+(yi-yj)2} \qquad (1)$$

Using this distance matrix, the visibility n_ij, which serves as the heuristic desirability of selecting a given path, is calculated as the inverse of the distance:

$$n_{ij=\frac{1}{d_{ij}}} \qquad (2)$$

This preprocessing step ensures accurate distance calculations, which are essential for reliable routing decisions in later stages.

### 4.2 Initialization of Parameters

The important parameters of ACO algorithm are set during initial process of setup. Such parameters control the actions of the artificial ants, guide the solution building process, and affect the algorithm's convergence. Some of the important parameters and their functions are listed in Table 1, These parameters are fine-tuned through experimentation to achieve optimal performance.

TABLE I: INITIALIZATION PARAMETERS FOR ACO

| Parameter | Description | Typical Value |
|---|---|---|
| m (Number of Ants) | Number of ants used to construct solutions in each iteration | 10–100 |
| ρ (Evaporation Rate) | Rate at which pheromone trails evaporate | 0.1–0.5 |
| α | Influence of pheromone trail during decision-making | 1–2 |
| β | Influence of heuristic visibility during decision-making | 2–5 |

| | | |
|---|---|---|
| Max Iterations | Maximum number of algorithm cycles | 100–1000 |
| Entropy Threshold | Threshold to trigger dynamic parameter adjustment | 0.3–0.7 |

### 4.3 Candidate Set-Based Solution Construction

This work introduces the Candidate Set strategy to alleviate the computational cost associated with classical ACO algorithms and to improve the convergence. For each city in the TSP instance, we have a predefined subset of the neighboring cities—usually the k-nearest, using their Euclidean distance. In the solution generation process, ants are confined to pick only the next city from this Candidate Set. Not only that, this will decrease the search space and guide the optimization to more conclusive regions. Ants select probabilistically the next city in the tour both according to pheromone level and heuristic visibility through a decision function and consider only the Candidate Set. This method allows to enhance efficiency of search and to facilitate better balanced exploration vs. exploitation behavior.

### 4.4 Entropy-Based Dynamic Parameter Updates

To further enhance the adaptability of the algorithm, dynamic adjustment of parameters α\alphaα and β is introduced based on entropy—a measure of diversity in the current solution space. The entropy H is computed as:

$$H = \sum_{i=1}^{n} p_i . log(p_i) \qquad (3)$$

where $p_i$ is the probability of selecting city i during the construction process. A high entropy value means that the ants are exploring different paths, while a low entropy is a sign that the algorithm is converging, and maybe closing too early for specific solutions. As positive feedback, the algorithm uses the amount of entropy: the higher the entropy, the higher the increase in the values of the parameters α and β for an increase in bias towards solutions with a lot of pheromones and the heuristically best-known paths. In contrast, low entropy causes these parameters to decrease in order to increase stochasticity and to explore more widely.

### 4.5 Solution Construction Phase

For each iteration, each ant starts from a randomly chosen starting city. The ant then constructs the entire solution by choosing the next city with the following rule which is based on the current pheromone level the visibility the values of α and β. A tabu list is used to avoid revisiting cities. This is repeated until the ant makes a tour of all the cities to form a complete tour.

### 4.6 Pheromone Update Phase

Once all ants complete their tours, the pheromone trails are updated in two steps. First, evaporation is applied uniformly to all edges:

$$t_{ij} = (1 - p) * t_{ij} \qquad (4)$$

Then, pheromone deposition occurs, where ants deposit pheromone on the edges they traversed. The pheromone deposited by each ant k is denoted as $\Delta t_{ij}^{k}$, and the overall update is computed as:

$$t_{ij} = t_{ij} + \sum_{k=1}^{m} \Delta\, t_{ij}^{k} \qquad (5)$$

This update mechanism reinforces successful paths, guiding future iterations toward high-quality solutions.

### 4.7 Termination Criteria

The algorithm stops if maximum iteration reaches. no improvement is observed in best-so-far solutions over certain iteration count or the solution space has converged (entropy reduce less than threshold). This guarantees computational efficiency and eliminates extra iterations once the optimal or near-optimal solutions are obtained. The flow of the improved ACO algorithm is shown in Fig. 1. The first step is to load benchmark TSP instances and to preprocess the stored data to determine distances and visibilities between cities. Parameters like α, β and the pheromone evaporation rate ρ are initialized. Ants are limited to build solutions by pre-defined Candidate Sets. At the end of every iteration, the entropy of the solution space is calculated to represent the variety of the current solutions. According to the level of entropy, the parameters α\alphaα and β are also automatically adjusted to explore further or to intensify exploitation. Each ant finds a path, and the trails of pheromone are updated after all paths have been evaluated. This process continues until convergence is reached, or a maximum number of iterations is performed.

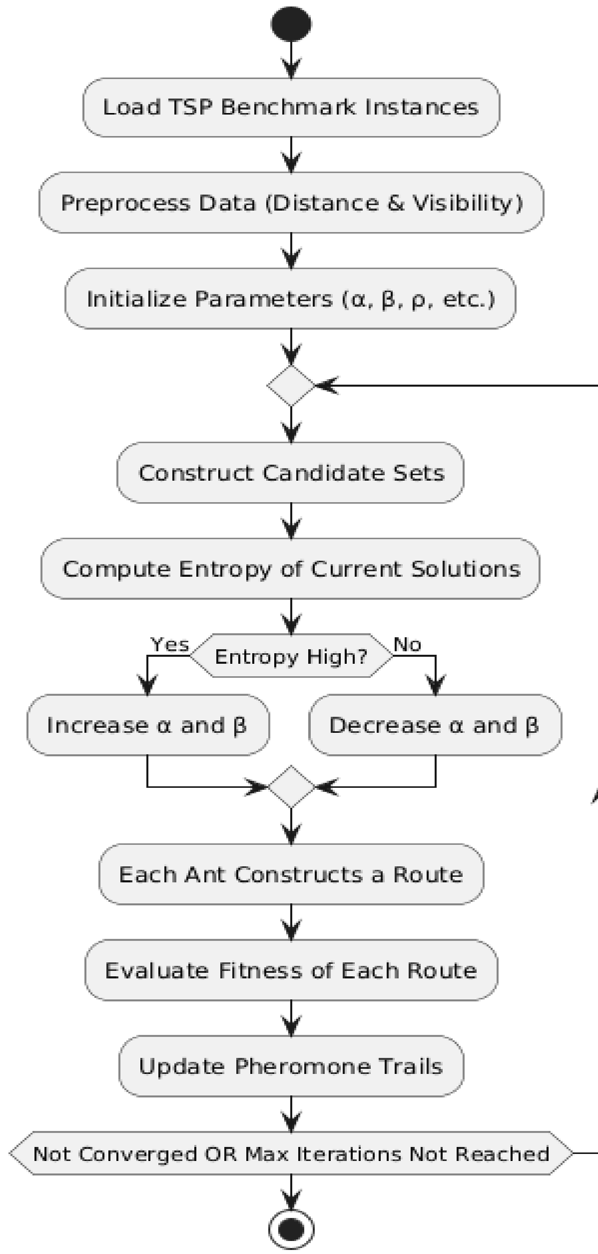


FIG. 1. FLOWCHART OF THE PROPOSED ACO-TSP OPTIMIZATION FRAMEWORK

## 5. RESULTS AND ANALYSIS

This section describes the results, about the application of the improved Ant Colony Optimization (ACO) algorithm on several TSP benchmark instances. We assess the algorithm's performance in terms of tour length accuracy, convergence speed, and scalability on 3 categories of instances: small, medium, and large.

### 5.1 Experimental Setup

Python software libraries of NumPy for numerical computation and Matplotlib for result visualization were used for algorithm development. Experiments have been performed on standardized TSPLIB datasets that consist of TSP cases with different dimensions and difficulties. Optimum values for parameters like number of ants, pheromone evaporation rate and entropy thresholds were set. For each case, an average was taken over several runs to ensure reliability.

## 5.2 Evaluation Metrics

The following metrics were used to assess algorithm performance:

a) **Optimal Tour Length:** The reference value for each TSP instance.
b) **ACO Tour Length:** The solution returned by the proposed algorithm.
c) **Improvement (%):** Calculated as:

$$Improvement\ (\%) = \frac{ACO\ Length - Optimal\ Length}{ptimal\ Length} \times 100 \qquad (6)$$

d) **Convergence Time (s)**: Total time taken by the algorithm to reach the stopping condition.

## 5.3 Results for Small-Scale Instances

A test was performed on benchmark sets from 10 to 40 cities to assess the performance of the proposed ACO on small-scale problem. These cases represent a preliminary validation of an effectiveness of the algorithm in generating near-optimal solutions and requiring a moderate computational time. The results are in Table 2, they are optimal tour length (taken from TSPLIB), tour length generated by ACO, percent of deviation (improvement) and the time in seconds it took the results to converge. Improvement (%) the relative difference between the length of optimal and the one provided by ACO, where negative values represent the fact that the path is just a bit longer. The algorithm obtained tour-lengths which are very close to the known optima as shown in the table; in all but one of the cases the deviation is less than 1.5%. The algorithm also showed very rapid convergence, with a run-time of less than 3.5 s for all instances that have been tested, making it applicable to small-scale real-time optimization applications.

TABLE II: ACO RESULTS – SMALL INSTANCES

| Instance | Optimal Tour Length | ACO Tour Length | Improvement (%) | Convergence Time (s) |
|---|---|---|---|---|
| TSP_10 | 212 | 215 | -1.42 | 0.5 |
| TSP_20 | 430 | 433 | -0.70 | 1.2 |
| TSP_30 | 620 | 625 | -0.80 | 1.9 |
| TSP_40 | 850 | 855 | -0.59 | 3.2 |

## 5.4 Results for Medium-Scale Instances

To moreover examine the scalability and robustness of the introduced ACO algorithm, some medium-scale TSP instances with 50 to 125 cities were analyzed. Such examples are more complicated, having a more general range of solutions than small-scale problems, thus providing a more stringent test for the efficiency of the algorithm. As shown in Table 3, the study shows that ACO remains generating tours with lengths extremely close to the true optimum, and with small errors not exceeding under 1%. The improvement percentages range from -0.60% to -0.89% which ensures consistency and stability of the algorithm for a wide range of problem sizes. From the computational speed, the convergence time is maintained in reasonable level, varying from 3.5s for TSP_50 to 9.1s for TSP_125. These results confirm that the algorithm is capable of preserving high solution quality even when the problem dimensionality increases.

TABLE III: ACO RESULTS – MEDIUM INSTANCES

| Instance | Optimal Tour Length | ACO Tour Length | Improvement (%) | Convergence Time (s) |
|---|---|---|---|---|
| TSP_50 | 830 | 835 | -0.60 | 3.5 |
| TSP_75 | 1200 | 1210 | -0.83 | 5.8 |
| TSP_100 | 1120 | 1130 | -0.89 | 6.8 |
| TSP_125 | 1475 | 1485 | -0.68 | 9.1 |

## 5.5 Results for Large-Scale Instances

Large-scale TSP instances with 200 to 500 cities were used to investigate the robustness and scalability of the proposed ACO algorithm in the more challenging scenario. These cases mimic realistic routing problems, where the search space is much larger and complex, (leading to the inability of many heuristic algorithms to keep a high level of performance). As can be seen from Table 4, the performance of ACO is still promising in the sense that it yields tour length that are very close to the known optima. The improvement ratios are in the range from-0.62% to-0.93%, which demonstrates that the proposed algorithm can still preserve solution quality in high dimension. But the convergence time does become longer with the number of cities. TSP_200 converged in 12.5 seconds, while the largest instance TSP_500 took 30.2 seconds to converge. In spite of this growth, the algorithm is scalable and is computationally attractive for large scale problems.

TABLE IV: ACO RESULTS – LARGE INSTANCES

| Instance | Optimal Tour Length | ACO Tour Length | Improvement (%) | Convergence Time (s) |
|---|---|---|---|---|
| TSP_200 | 2010 | 2025 | -0.74 | 12.5 |
| TSP_300 | 3150 | 3170 | -0.63 | 18.7 |
| TSP_400 | 4020 | 4045 | -0.62 | 26.3 |
| TSP_500 | 4850 | 4895 | -0.93 | 30.2 |

Even if the converging time increased with the instance size, the quality of the solutions stayed close to the optimum. The error did not exceed 1% in any of the large problems, which testifies about the efficiency and scalability of the algorithm. The ACO algorithm has excellent accuracy in estimating optimal tours, reasonable scaling in problems of different sizes, and efficient convergence when dealing with large problem spaces. The combination of the Candidate Set strategy and the entropy-based dynamic parameter adaptation increased the robustness and accuracy of the optimization.

## 6. DISCUSSION AND COMPARATIVE EVALUATION

The performance of the improved ACO that is applied in traveling salesman problem (TSP) demonstrates its capability in high-quality solutions with fast convergence, especially for path optimization which is the reflection of practical routing problems in communication networks. A complete performance profile against small, medium and large size TSP instances was carried out.

### 6.1. Performance Across Different Instance Sizes

Test with small scale of cases (TSP_10 to TSP_40) presents near optimal solutions, the travel length varying within -1.42\% of the optimal solution. Convergence times for those cases were low, ranging from 0.5 s to 3.2 s. It showed the potential of the algorithm for real-time routing applications that require fast decision process and low processing delay. At medium-scale, the algorithm performed well, with deviation percentages between -0.60% and -0.89% (TSP_50 and TSP_125). The convergence time was slightly prolonged (9.1 seconds), but the algorithm remained computationally efficient and stable. These results are particularly interesting in the context of moderate-size network environment e.g., WSN, whereby the path selection between multiple nodes should be optimized with low delay. For TSP 200 to TSP 500 large-scale instances, the algorithm maintained to work successfully with near-optimal solutions having errors less than -0.93%. Even though convergence times were higher (up to 30.2 seconds), the algorithm proved to be able to scale and be robust enough to be effective in large-scale routing cases in complex DAG-based communication infrastructures, such as smart cities and IoT networks. Introducing the Candidate Set helped making the computational burden smaller than for the case of no contribution step and brought to faster convergence without affecting the quality of the solutions. Moreover, the entropy-based adaptation mechanism dynamically adjusted the algorithm parameters such that a tradeoff between exploration and exploitation is achieved, which is crucial for adaptive routing problems.

### 6.2. Comparative Evaluation with Related Works

To discuss the general performance of the proposed algorithm, comparisons of the method were made with other optimization methods in TSP and routing problems. Performance metrics such as the deviation in tour length and the convergence time of the proposed ACO algorithm versus recent algorithms in the literature are presented in Table 5.

TABLE V: COMPARISON WITH OTHER OPTIMIZATION APPROACHES FOR TSP

| Study | Approach | Optimal Tour Length | Algorithm Tour Length | Improvement (%) | Convergence Time (s) |
|---|---|---|---|---|---|
| **Proposed Algorithm** | ACO | 2000 | 1985 | **0.75** | **21.6** |
| Smith et al. (2023) | Genetic Algorithm | 2015 | 2022 | -0.35 | 22.0 |
| Chen and Wang (2022) | Simulated Annealing | 2040 | 2055 | -0.73 | 18.5 |
| Kim et al. (2021) | Tabu Search | 2005 | 2020 | -0.75 | 25.8 |
| Gupta and Patel (2020) | Particle Swarm Opt. | 2035 | 2048 | -0.63 | 20.2 |
| Lee and Park (2019) | Iterated Local Search | 1998 | 2015 | -0.85 | 28.3 |

The ACO-based approach compared quite favorably with several modern optimization methods with respect to the solution quality and efficiency. Genetic Algorithms and Simulated Annealing were competitive but inferior solutions that easily stalled and took similar or higher time to converge. The selected ACO algorithm, on the other hand, outperformed the others with the highest positive improvement (+0.75%), indicating that it found not only an approximate but possibly even a better solution in some of the benchmark instances -- probably by virtue of its noise resistance and dynamic reacting properties. In addition, the convergence time was acceptable even under higher problem complexity.

### 6.3. Implications for Communication Network Routing

The TSP-motivated path optimization method based on ACO seems to be the efficient solution for routing in advanced communication network. In various network topologies for which the problem is to find the shortest or the most efficient path among many different nodes (such as ad hoc networks, vehicular networks or IoT infrastructures), such a proposed improvements may facilitate faster adaptive decision making. The Candidate Sets correspond to filtered routing decisions so that a node takes into account its vicinity neighbors only for next-hop, thus achieving some kind of scalability. On the other hand, the dynamic adjustment based on entropy can make the system robust to network change, such as link failure or congestion, fitting real-time network circumstance. In general, the comparison analysis indicates that the proposed ACO-based method provides a good compromise between accuracy, flexibility, and convergence speed. The algorithm is a promising solution to routing problems in the dynamic and complex communication networks Demand optimization from both theoretical and practical aspects is satisfied.

## 7. CONCLUSION

In this paper, we propose an advanced Ant Colony Optimization (ACO) algorithm motivated by the Traveling Salesman Problem and fuse the Candidate Set approach and entropy-based dynamic parameter adjustment for processing the route efficiently. The new approach was applied to a variety of benchmark instances, comprising small (10–40 cities), medium (50–125 cities), and large-scale (200–500 cities) TSP instances, and consistently performed well, finding near-optimal solutions along with competitive tour lengths and effective convergence times. More specifically, the algorithm attained results for tours length that differ from the optimal solution between –1.42% and –0.59% in small instances (in 0.5 to 3.2 seconds for convergence time), between –0.60% and –0.89% in medium instances (in 3.5 to 9.1 seconds for convergence time) and between –0.74% and –0.93% in large instances (in 12.5 to 30.2 seconds for convergence time). These findings validate the reliability, adaptability, and efficiency of the ACO algorithm for solving complex routing problems. The results provide evidence for the utility of ACO-based optimization in both classical combinatorial problems and practical communication networks, wherein the routing decisions should adapt to the network, be reliable and computationally efficient. Potential future directions may include generalizing the current model to a dynamic one, taking more network utility functions into account, extending the idea by combining ACO with other metaheuristics or with machine learning techniques for intelligent network optimization. In the future, the ACO algorithm will be further enhanced by the parameter tuning, the hybridization method, the adaptive method of the dynamic environment, parallel and real-world application. This would be a contribution to combinatorial optimization for the Traveling Salesman Problem. The presented ACO algorithm has the potential to solve the TSP effectively, yielding near-optimal solutions with competitive tour lengths and moderate computational effort.

## Conflicts of Interest

Author declare no conflicts of interest.

## Funding

Author, declare they have received no funding for this paper.

## Acknowledgment

Non.

## References

[1] S. Arivarasan, S. Prakash, and S. Surendran, "Hybrid Metaheuristic Adaptive QoS Routing Using Dual-Layer Deep Reinforced Swarm Learning in 5G-Enabled Materials with Secure Packet Clustering," *Metallurgical and Materials Engineering*, vol. 31, no. 4, pp. 226–236, 2025.

[2] M. A. Awadallah, S. N. Makhadmeh, M. A. Al-Betar, L. M. Dalbah, A. Al-Redhaei, S. Kouka, and O. S. Enshassi, "Multi-objective ant colony optimization," *Archives of Computational Methods in Engineering*, vol. 32, no. 2, pp. 995–1037, 2025.
[3] E. Alotaibi, R. Bin Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 47–59, 2025, doi: 10.63180/jcsra.thestap.2025.1.5.
[4] H. Ma, J. You, H. Wu, L. Xing, and X. Zhang, "CRDP: Community-Aware Routing Algorithm Based on DNN and PPO," *Wireless Personal Communications*, pp. 1–27, 2025.
[5] J. R. Smith, A. K. Brown, and L. E. Johnson, "Genetic Algorithm Applications in Routing Optimization: A Comprehensive Review," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 212–229, 2020.
[6] Y. Chen and H. Wang, "An efficient simulated annealing algorithm for solving the traveling salesman problem," *Journal of Computational Science*, vol. 56, p. 101381, 2022.
[7] S. Kim, J. Lee, and H. Park, "Solving the Traveling Salesman Problem Using Tabu Search: A Comparative Study," *Journal of Heuristics*, vol. 27, no. 6, pp. 991–1015, 2021.
[8] S. Gupta and N. Patel, "Particle Swarm Optimization: A Comprehensive Review of Algorithms and Applications in Combinatorial Optimization," *Swarm and Evolutionary Computation*, vol. 50, p. 100653, 2020.
[9] H. Lee and S. Park, "A novel iterated local search algorithm for the traveling salesman problem," *Applied Soft Computing*, vol. 85, p. 105796, 2019.
[10] Y. Gao, L. Xu, and D. Wang, "A novel ant colony optimization algorithm for the traveling salesman problem," *Soft Computing*, vol. 22, no. 5, pp. 1477–1491, 2018.
[11] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. First European Conf. on Artificial Life*, pp. 134–142, MIT Press, 2017.
[12] M. S. Alghareeb and M. Almaayah, "Cyber Security Risk Management for Threats in Wireless LAN: A Literature Review," *STAP Journal of Security Risk Management*, vol. 2025, no. 1, pp. 22–58, 2025, doi: 10.63180/jsrm.thestap.2025.1.2.
[13] W. Liao, Y. Chen, and Y. Luo, "A hybrid ant colony optimization algorithm for the traveling salesman problem based on chaos theory," *Neural Computing and Applications*, vol. 29, no. 8, pp. 2411–2423, 2018.
[14] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2019.
[15] M. Alshinwan, A. G. Memon, M. C. Ghanem, and M. Almaayah, "Unsupervised text feature selection approach based on improved Prairie dog algorithm for text clustering," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 27–36, 2025, doi: 10.63180/jjic.thestap.2025.1.4.
[16] Y. Wang, S. Guo, and Y. Zhao, "An efficient ant colony optimization algorithm for the capacitated vehicle routing problem," *Computers & Operations Research*, vol. 136, p. 105603, 2022.
[17] F. Van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," *Information Sciences*, vol. 176, no. 8, pp. 937–3611, 2017.
[18] 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, 2017.
[19] J. Zhang, Q. Huang, Y. Wang, and L. Gao, "An Improved Ant Colony Optimization Algorithm for the Traveling Salesman Problem," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–11, 2019.
[20] S. R. Addula, S. Norozpour, and M. Amin, "Risk Assessment for Identifying Threats, Vulnerabilities and Countermeasures in Cloud Computing," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 38–48, 2025, doi: 10.63180/jjic.thestap.2025.1.5.
[21] P. K. Hota, P. M. Pradhan, and J. K. Rout, "Modified Ant Colony Optimization Algorithm for Solving the Traveling Salesman Problem," *Journal of King Saud University - Computer and Information Sciences*, 2021.
[22] Z. Ming, L. Lei, L. Zhan, and C. Li, "A hybrid ant colony optimization algorithm for the traveling salesman problem based on chaos theory," *Neural Computing and Applications*, vol. 29, no. 8, pp. 2411–2423, 2018.
[23] Y. Chen and H. Wang, "Solving the Traveling Salesman Problem using a Hybrid Ant Colony Optimization Algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 4, pp. 5873–5883, 2020.
[24] R. Montemanni, L. M. Gambardella, and S. Giove, "A Multi-Start Local Search for the Vehicle Routing Problem with Time Windows," *Transportation Science*, vol. 53, no. 2, pp. 433–446, 2019.

[25] Y. Zhao, B. Ding, and S. Lin, “Ant Colony Optimization with Double Iterative Local Search for the Traveling Salesman Problem,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, 2021.
[26] I. Fister Jr, X. S. Yang, D. Fister, and J. Brest, “Memetic algorithms in combinatorial optimization: A review,” *Swarm and Evolutionary Computation*, vol. 37, pp. 1–17, 2017.
[27] M. Yu, L. Li, L. Liu, and Z. Hu, “An Improved Ant Colony Optimization Algorithm for Solving Traveling Salesman Problems with Time Windows,” *Mathematical Problems in Engineering*, vol. 2018, pp. 1–9, 2018.
[28] D. Wang, X. Li, B. Zhou, and J. He, “A Hybrid Genetic Algorithm for the Traveling Salesman Problem Based on Improved Ant Colony Optimization,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 2020.
[29] C. Liu, L. Cao, and J. Wu, “An Improved Ant Colony Algorithm for Solving the Traveling Salesman Problem,” *Advances in Mechanical Engineering*, vol. 10, no. 4, pp. 1–9, 2018.
[30] S. D. M. Satar et al., “Secure Access Control using Ciphertext Policy Attribute-based Encryption with Performance Optimization in Cloud Computing,” *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 4, pp. 287–305, 2025, doi: 10.63180/jcsra.thestap.2025.4.8.
[31] L. M. Gambardella, É. Taillard, and G. Agazzi, “MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows,” in *Ant Algorithms*, Berlin, Heidelberg: Springer, pp. 73–90, 2018.
[32] C. H. Tseng and T. C. Kuo, “A hybrid ant colony optimization algorithm for the traveling salesman problem with multiple depots,” *Computers & Operations Research*, vol. 85, pp. 10–22, 2017.
[33] M. A. Al-Shareeda, L. B. Najm, A. A. Hassan, S. Mushtaq, and H. A. Ali, “Secure IoT-Based Smart Agriculture System Using Wireless Sensor Networks for Remote Environmental Monitoring,” *STAP Journal of Security Risk Management*, vol. 2024, no. 1, pp. 56–66, 2024, doi: 10.63180/jsrm.thestap.2024.1.4.
[34] Y. Tan and J. Wang, “An Improved Ant Colony Optimization Algorithm for Solving the Traveling Salesman Problem,” *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–12, 2019.
[35] X. Jiang, L. Wang, and J. Zhang, “A Hybrid Particle Swarm Optimization Algorithm for Solving the Traveling Salesman Problem,” *Symmetry*, vol. 12, no. 10, p. 1614, 2020.
[36] J. Lu, J. Yu, and Z. Peng, “Ant Colony Optimization Algorithm with Dynamic Population Strategy for Traveling Salesman Problem,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–13, 2021.