Ant colony optimization for traveling salesman problem: A review

Abstract

The traveling salesman problem (TSP) is a fundamental combinatorial optimization problem with applications in resource management, logistics, and communications. In order to address TSP and its differences, this paper discusses developments in Ant Colony Optimization (ACO), a biologically inspired algorithm. Inspired by the foraging activity of ants, ACO's decentralized and recursive methodology has proven successful in solving difficult routing problems. ACO's scalability, convergence speed, and solution quality have been greatly enhanced over time through innovations including hybridization with algorithms such as Firefly, genetic algorithms, parallel computing frameworks, and adaptation mechanisms. These developments have given the ACO the flexibility and efficiency to handle dynamic situations, such as real-time vehicle guidance and underwater navigation. Despite its progress, issues remain such as scalability in resource-limited contexts, processing overhead, and reliance on parameter modification. This work summarizes current developments in ACO, noting how revolutionary the TSP solution is, pointing out its drawbacks, and suggesting areas for further study. With the help of cutting-edge technologies such as machine learning and quantum computing, ACO has huge potential to progressively address challenging real-world problems. This review provides a comprehensive framework for developing uses of ACOs and reaffirms their status as a key component of improvement research.

Keywords- Traveling Salesman Problem (TSP), Ant Colony Optimization (ACO), Metaheuristic Algorithms, Dynamic Routing, Hybrid Optimization Techniques, Parallel Computing in Optimization, Real-World Applications of ACO.

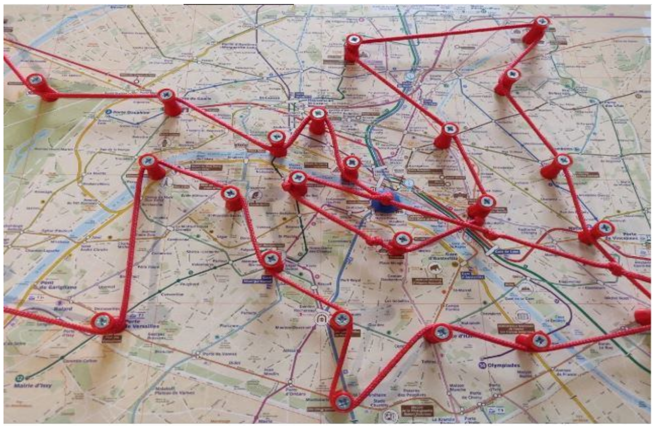
Introduction

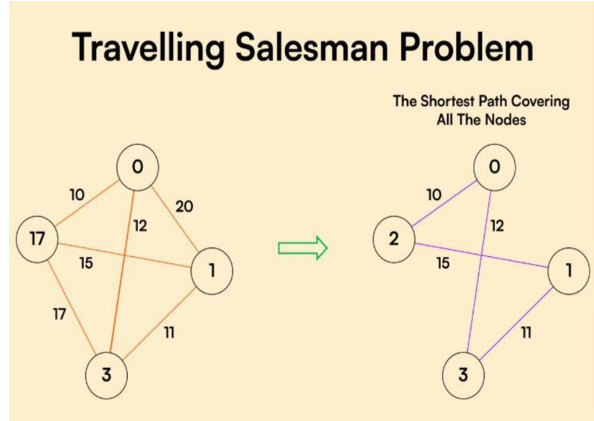
In combinatorial optimization, the Travel Salesman Problem (TSP) is one of the most difficult and well-researched problems. It is an NP-hard problem that finds the fastest way for a travel salesman to a set of cities precisely once and then back to the starting point. The practical importance of this topic extends to industries where efficient routing and allocation of resources is critical, including manufacturing, bioinformatics, logistics, and telecommunications [1]. Although TSP is a straightforward formula, its computational complexity requires the application of heuristics and methods to find near-optimal solutions. Ant colony optimization (ACO) is one such technique that has become very popular [2], [3]. Developed by [1]in the 1990s, ACO is based on the foraging behavior of ants, which use pheromone trails to collectively explore and optimize routes. This biologically inspired method has shown great potential in tackling a range of optimization problems, especially TSP and its variations. However, traditional CCOs suffer from some drawbacks such as slow convergence and weakness in the face of local optimal conditions. The pheromone mutation and reconfiguration techniques pioneered by [4],[5] were among the early developments that effectively mitigated some of these problems. Hybrid strategies have been more effective tools for improving CCO performance in recent years.

For example in [6] combined ACO with the Firefly algorithm, achieving higher convergence time and solution quality, while [7] used ACO with mutation techniques to optimize DNA sequencing workloads. ACO can now efficiently handle large-scale problems thanks to parallelism, which has increased its application. Studies by [8],[2]have shown how parallel ACO can be used in real-world applications because it reduces computation time without sacrificing solution accuracy. Adaptive mechanisms have improved the flexibility of ACO in dynamic and constrained situations. Examples of such mechanisms are the dynamic parameter tuning frameworks proposed by [9],[10]. These advances have demonstrated that ACO is a highly flexible algorithm that can handle the challenges of truck routing, logistics, and other real-world applications [11],[12]. Even with these advances, some limitations still exist. According to [13],[14], hybrid models often lead to increased computational cost, and parameter tuning remains a major hurdle to maximizing ACO performance. Future studies should address the scalability and complexity of TSP by leveraging quantum computing, machine learning-based parameter automation, and lightweight hybrid frameworks [15],[16],[17]. Addressing these issues would help ACO maintain its position as a core component of optimization research. This review study examines the evolution of ACO and its applications to TSP, identifies its limitations, and outlines possible avenues for innovation. By combining the findings of previous research, this work helps to fill knowledge gaps and encourage further progress in this important area of improvement studies.

Theoretical Framework

The Traveling Salesman Problem (TSP) It is a well-studied reference problem in combinatorial optimization due to its practical importance and computational difficulty. The goal is to determine the quickest way for a seller to travel to a group of cities at once and then return to the starting location.The complexity of solving this seemingly simple task increases with the number of cities, making it NP-hard. TSP is used in a variety of fields where efficient resource allocation and direction is essential, including bioinformatics, robotics, telecommunications, and logistics [1]. Ant colony optimization (ACO) has been a prominent approach among heuristic and metaheuristic approaches due to the inability of traditional methods to scale well to large problem sizes [4],[13] as shown in Figures 1 and 2.





Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO),It's a biologically inspired metaheuristic algorithm that mimics the way ants eat in the wild. In their natural environment, ants use pheromones—chemicals that signal routes between their colony and food sources—for indirect communication. As more ants use shorter, more effective pheromone trails, it gradually strengthens the colony and leads it in th cial ar using pl itera ACO cc nown i

Figure 1: The Traveling Salesman Problem (TSP) is represented on a map with red pathways signifying the shortest path and nodes representing cities.

Figure 2: A condensed TSP graph that effectively connects every node using the shortest path.

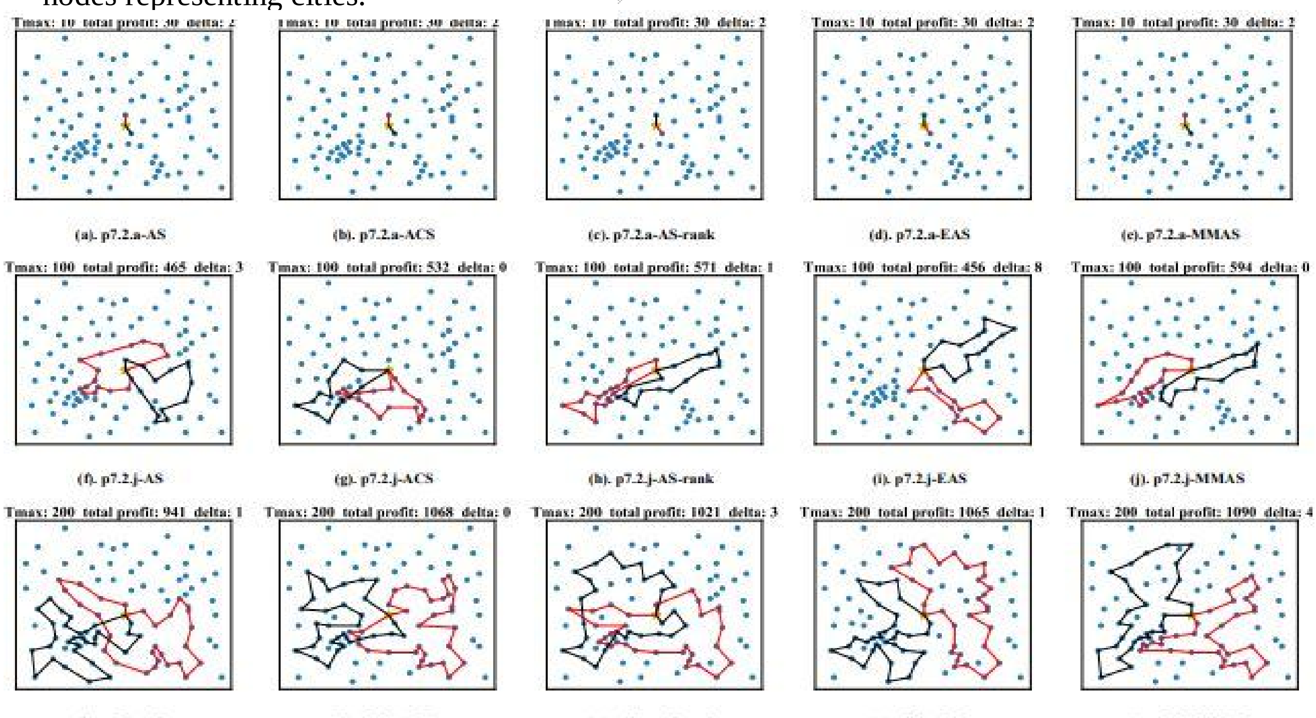


Figure 3: The Five ACOs Discovered Two Ideal Routes[13].

In the ACO framework, artificial ants move around a graphical representation of the problem space, choosing paths in a probabilistic manner by combining problem-specific heuristic values and pheromone intensity. Three basic elements are the foundation of an ACO:

Pheromone Trails

The desirability of paths in the solution space is represented by pheromone paths. On their travels, artificial ants leave behind virtual pheromones, the amount of which often correlates with the titer of the treatment detected. Pheromones gradually dissipate, which reduces their effectiveness and promotes research into various methods. This evaporation maintains the diversity of the search process and prevents the algorithm from stumbling upon less-than-optimal solutions. Improved pheromone updating mechanisms have been used in studies such as those by [12] and [4],[5]to improve solution convergence and prevent premature stagnation. [7]used mutational techniques to improve pheromone updates, resulting in higher results in domain-specific applications such as DNA sequencing.

Heuristic Information

Ants are given more guidance when choosing a path through heuristic knowledge. Shorter distances are preferred in the context of TSP, as indicative values are usually determined from distances between cities. Ants are able to balance local and global exploration efficiently using this information along with pheromone levels to calculate metamorphosis probabilities. According to research by [9]and [10], adaptive heuristic weight ad

justments are essential to enhance algorithm performance in dynamic problem contexts.

Exploration vs. Exploitation

An ACO's ability to balance exploitation (improving existing high-quality solutions) and exploration (finding new solutions) is a critical element. Pheromone evaporation rates and the effect of signal information are two examples of parameters that regulate this balance. While over-exploration can slow the convergence process, over-exploitation can lead to premature convergence on poor solutions. According to [2], improving these factors can significantly improve the ability of ACOs to solve complex optimization problems. In order to improve conflicting goals in routing issues, [14]created a multi-objective ACO framework that adaptively modified exploration and exploitation.

Applications of ACO in Optimization

ACO is very good at tackling complex optimization problems, especially TSP and its variations, because of its decentralized structure, flexibility, and iterative optimization. Studies by [2] and [8] have demonstrated the effectiveness and scalability of ACO in practical applications such as supply chain management and logistics. [13] used adaptive exploration and ACO techniques to balance path stability and profits in a team routing problem (TOP). The utility of the algorithm in guiding unmanned underwater vehicles was demonstrated by[14], who addressed issues including resource allocation and dynamic constraints.

Advancements in ACO

By incorporating cutting-edge computing methods and hybridizing with other algorithms, ACO has seen tremendous development over time. ACO was combined with the Firefly algorithm by [6] to improve convergence rates and adjust parameters dynamically. By combining ACO with genetic algorithms,[18] increased the power and diversity of their solutions. Applications of parallel computing, including those introduced by [8], have greatly enhanced the capabilities of ACO and made it possible to compute large-scale problems more quickly. ACO has become a mainstay in optimization research by taking advantage of these advances, successfully dealing with both theoretical problems and real-world applications. Its continued progress highlights its adaptability and importance in resolving a wide range of consensual and practical issues.

Metaheuristic Algorithms in TSP

Because metaheuristic algorithms can effectively explore and exploit enormous, complicated solution spaces, they have become essential tools for solving the Traveling Salesman Problem (TSP). Metaheuristics use probabilistic techniques to iteratively improve solutions, in contrast to deterministic algorithms that depend on exhaustive search or preset rules. Its adaptability to various problem constraints and its ability to escape local optima make it particularly useful for NP-hard problems such as TSP [1].

Key Features of Metaheuristic Algorithms

The power, scalability and adaptability of meta-algorithms define it. To traverse the solution space, they use strategies including population-based search, heuristic-based routing, and random selection. These properties allow metaphysics to achieve a balance between exploitation (intensifying searches in promising locations) and exploration (searching in new areas). Some well-known meta-methods of TSP annealing (SA), particle swarm optimization (PSO), genetic algorithms (GA), and ant colony optimization (ACO) are simulated [2], [9].

Ant Colony Optimization (ACO) as a Metaheuristic

An important metaphysical factor to solve the TSP problem is ACO, which was inspired by the feeding habits of ants. Large solution spaces can be explored efficiently thanks to the decentralized search process and pheromone-based learning mechanism. ACO also has some disadvantages, such as the possibility of early convergence and its sensitivity to changes in parameters. To address these issues, researchers have created hybrid algorithms that incorporate the benefits of multiple metaheuristics techniques [7],[13].

Hybridization with Other Metaheuristics

By introducing complementary technologies, hybrid MTs improve ACO performance. The ACO algorithm and Firefly (FA) were combined into a hybrid algorithm by [6], where the FA continuously adjusted ACO parameters to avoid stagnation and accelerate convergence. In large-scale TSP cases, this integration reduced computation time and improved solution quality. Similar to this, [7] improved search diversity and achieved better results in DNA sequencing tasks designed by TSP by incorporating mutational techniques into the ACO pheromone updating process. The drawbacks of ACO have also been successfully addressed by combining it with genetic algorithms (GA). [18] proposed a hybrid strategy in which refined solutions were optimized by ACO and the solution space was diversified using crossover and mutation operators in GA. Convergence rates and solution power are enhanced by this synergy, especially in dynamic TSP settings. To improve parameter tuning and adapt to changing problem constraints, Particle Swarm Optimization (PSO) has also been used with ACO [17],[12].

Advantages of Hybrid Metaheuristics

There are various advantages to metaheuristic hybridization:

1. Increased convergence speed: The number of iterations needed to produce high-quality solutions is reduced when ACO is combined with fast convergence algorithms such as PSO [14].
2. Better solution quality: To avoid local optimization and investigate different regions of the solution space, hybrid algorithms take advantage of complementary [6].
3. Robustness in dynamic environments: Adaptive approaches allow hybrid algorithms to continue working even when faced with various constraints, including changing resource availability or flight times [12],[11].

Applications of Hybrid Metaheuristics

Several real-world applications of TSP have demonstrated the effectiveness of hybrid methanes. For example, [13] balanced path stability and profit maximization using a hybrid ACO-FA algorithm to solve the team routing problem (TOP). [14] Optimizing flight time and resource allocation in unmanned underwater vehicle routing using a multi-objective hybrid ACO framework. These experiments demonstrate how hybrid metaheuristics can deal with a variety of challenging optimization problems.

Optimization Techniques for TSP

The performance of Ant Colony Optimization (ACO) in solving the Traveling Salesman Problem (TSP) has improved significantly over time through advances in optimization techniques. These developments address fundamental issues including preventing premature convergence, enhancing the level of solutions, and dynamically adapting to problem boundaries. From simple problems to complex real-world applications, ACO's cutting-edge capabilities have allowed it to successfully handle a wide range of TSP variables[2].

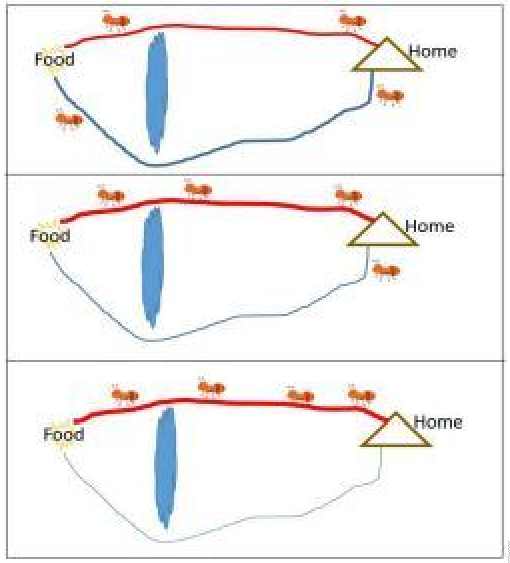


Figure 4: How ants locate food in the wild [2].

Pheromone-Based Strategies

The key to the effectiveness of an ACO is the administration of pheromones. Artificial ants leave behind pheromone trails that guide further search iterations, encouraging the search for better solutions while discouraging the pursuit of less useful methods. However, fixed pheromone updating techniques may cause suboptimal solutions to converge too early. Improved pheromone-based tactics have been created to solve this problem:

- Mutation-enhanced pheromone updates, introduced by[4], provide more diversity in the search and stasis-avoidance process. The algorithm can avoid local optima and search for alternative routes by regularly changing pheromone levels.

- In [8] suggested resetting pheromone levels at regular intervals while conducting research. This method preserves ACO's exploratory capabilities over the long term by preventing the algorithm from being too biased toward a single solution path.

- Quality-based pheromone deposition: In order to ensure that high-performance solutions receive further enhancement, [7] Tailor-made deposition of pheromones in proportion to solution quality. The ability of the algorithm to recognize the best routes and use them efficiently is enhanced by this optimization.

Local Search Methods

ACO is frequently used in conjunction with local search strategies to improve results within the communities of a given search space. These techniques work on the principle of incremental improvement, which involves making small modifications to an already existing solution to increase its quality. The ability of ACO to converge to optimal or near-optimal solutions is improved by incorporating local search:

- 2-opt and 3-opt heuristics:[19] showed how to optimize trajectories in TSP situations by incorporating 3-opt moves into ACO. This method systematically removes crossings from the route, increasing the overall length of the trip.

- Neighborhood exploration: [12] used adaptive neighborhood search strategies in addition to ACO, which allowed the algorithm to investigate alternative solutions that were close to the optimal solution at the moment. This integration improved convergence rates while requiring less computing power.

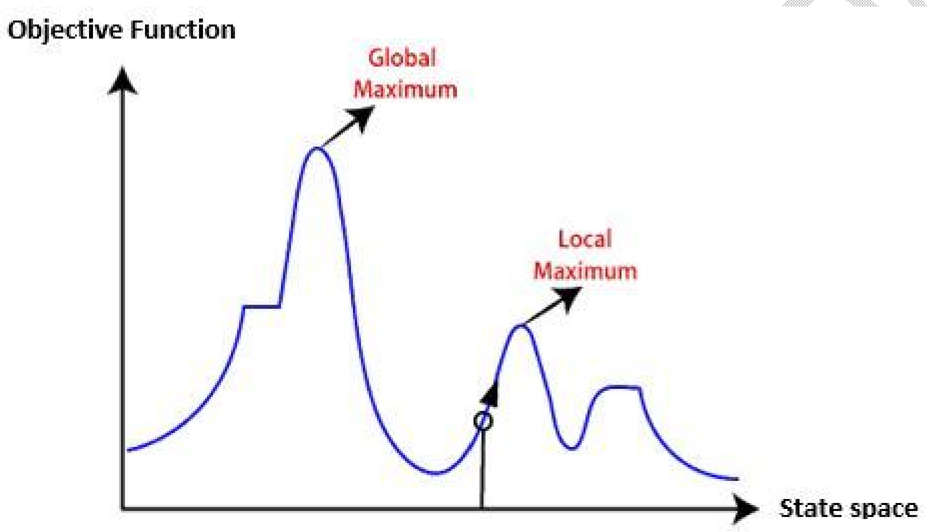


Figure 5:Height in a one-dimensional space scene is where the objective function exists.

Adaptive Parameter Tuning

Adaptability is one of the main advantages of an ACO, and new advances in adaptive parameter tuning have made it even more flexible. The performance of conventional ACO may be limited in dynamic problem scenarios due to its reliance on fixed factors, such as fixed pheromone evaporation rates and heuristic weights. Adaptive parameter tuning effectively adjusts these parameters according to the problem condition and algorithm performance:

• **Dynamic evaporation rates:**[9] presented a system that adjusts pheromone evaporation rates according to the solution quality. Early iterations use lower evaporation rates to enhance exploration, while later iterations use higher rates to focus on exploiting interesting paths.

•**Heuristic Weight Adjustment**: [10]Proposed heuristic weighting methods that adjust the relative importance of heuristic information according to mission parameters, such as time limits and city density. This ensures that the method will continue to work well in a variety of TSP conditions.

Hybrid Models in ACO

Overcoming the limitations has become largely possible thanks to hybrid models that combine ACO with free algorithms. The efficiency of combining ACO with the Firefly algorithm was demonstrated by [6], where the FA actively adjusts the ACO settings to avoid local optimum stagnation. ACO was combined with genetic algorithms by [18] to enhance exploration capabilities and ensure a variety of solution methods. In order to improve pheromone pathways for applications such as DNA sequencing, [7] Integrating mutation techniques into ACO. In addition to improving the quality of solutions, these hybrid models increase the scalability of ACO, qualifying it to address optimization problems with large constraints and size.

Dynamic Routing Applications

Important issues arise from dynamic routing problems, when environmental factors such as traffic, time, and resource availability change over time. Due to its versatility, an ACO is the perfect choice for these issues.[13]presented strategies to balance path stability and revenue by applying ACO to the team routing problem (TOP). A non-supervised sorting ACO (NSMOACO) was created by [14] in a similar way to maximize flight time and resource utilization in guiding unmanned underwater vehicles. These experiments demonstrate how well ACO can handle real-world dynamic routing problems.

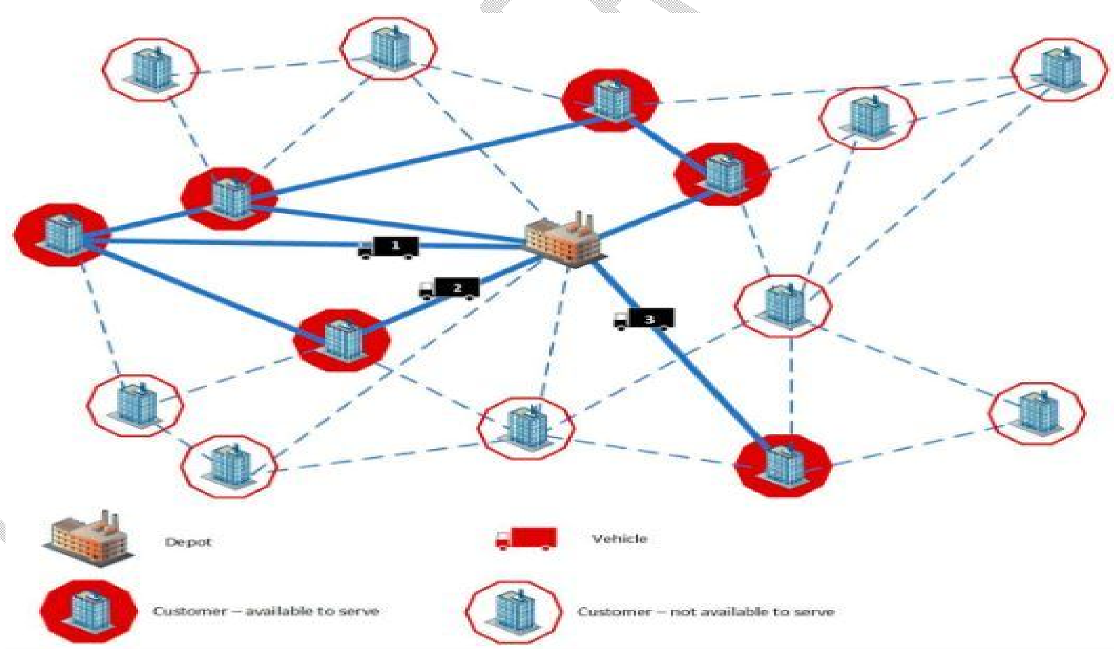


Figure 6: According to the general rule $\rho\rho$, the expected consumer availability difficulty is the initial circumstance (initial state), where only six of fifteen clients are available for the service.

Parallel Computing in ACO

Ant colony optimization (ACO) has been transformed by parallel computing, which greatly increases its scalability and efficiency. When used for large-scale problems, traditional ACO algorithms often face computational difficulties due to their stringent pheromone updating procedures and the requirement to evaluate a large number of possible solutions. Researchers have overcome these limitations through the use of parallel computing, which allows ACO to solve more complex and large-scale problems in an acceptable amount of time. ACO is an effective tool for industrial-scale optimization problems because of its incorporation of parallelism, which not only accelerates convergence but also maintains or improves the quality of the solution.

The Need for Parallelism in ACO

Particularly for large-scale cases of the Traveling Salesman Problem (TSP) and its variations, the computing complexity of ACO grows exponentially with the problem size. This intricacy results from:

1. Iterative solution construction: Each artificial ant builds an exhaustive solution in each iteration, which requires a large amount of computing power.

2. Pheromone updates: It takes a lot of resources to update the pheromone trails based on all the solutions generated.

3. Exploration of the search space: Since there are many possible paths in large-scale problems, exhaustive exploration is not practical.

Applications of Parallel ACO

Several industrial-scale optimization challenges have seen the successful application of parallel ACO:

1. Supply chain management and logistics

In order to optimize delivery routes in large-scale logistics networks taking into account time windows and vehicle capacities, [2] used parallel ACO. This technology was able to easily handle datasets containing thousands of nodes thanks to the parallel framework, which reduced the overall computing time.

2. Design of communications networks

Parallel ACO has been used by [8]in the design of communications networks, where capacity utilization and latency minimization are crucial. Evaluation of potential network configurations was accelerated through parallel execution, producing optimal designs in a fraction of the time needed for classic ACO.

3. Dynamic resource allocation

[13]provided an example of how parallel ACO can be used in situations involving dynamic resource allocation, such as assigning real-time tasks to cloud servers. The algorithm actively adjusts to changing resource availability and task priorities by balancing solution production and pheromone updates.

The advantages of parallel ACO

ACO's incorporation of parallel computing provides several important benefits.

1. Faster convergence: ACO can converge to optimal solutions more quickly thanks to parallel execution, which significantly reduces the time needed for each iteration[8].
2. Scalability: ACO is suitable for industrial-level challenges where parallel frameworks can manage large data sets and complex problem instances [2].
3. Improved exploration: Parallel ACO maintains search diversity by dividing the computation among many processors, reducing the possibility of premature convergence [13].

Literature Review

Sciannaet al.,[1] A modified version of the ant colony optimization (ACO) algorithm, AddACO, is presented in the study to overcome the shortcomings of traditional ACO methods for the traveling salesman problem (TSP). The three algorithmic variations involving pheromone trajectories, unpredictability, and inertia, as well as the linear convex composition method for decision making, are important innovations. The AddACO variations outperform classical ACO systems in terms of efficiency and heuristic capabilities, as experimental results on medium- and large-scale TSP examples reveal. It also increases the solution quality, computation time, and convergence speed.

Baydogmus et al,[2] This study focuses on using parallel ACOs to deal with the increasing computational complexity of TSPs as the number of cities increases. While maintaining the quality of the solution, the parallelization technique significantly reduces the execution time. By using parallel processing, Baydogmus showed that while more colonies speed up the optimization process, they also increase the time complexity. Parallel ACOs can be very effective at solving large-scale TSPs, according to the paper, making them suitable for practical uses such as transportation and logistics.

Dou et al., [3] proposed to extend the multiple travel vendor problem with visitation constraints (VCMTSP) to include ACOs. This strategy complicates the problem by taking into account vendor accessibility constraints. To solve the multiple travel vendor problem with hub cities, Chen (2024) modified the ant colony system (ACS) so that many vendors visit the cities with the highest demand. This approach balances the length of each agent's route while minimizing travel cost. Both studies demonstrate how ACOs can adapt to complex routing situations with additional constraints.

Ratanavilisagul et al., [4]In this paper, a mutation technique applied to pheromones is used to improve the ant colony optimization (ACO) for the traveling salesman problem (TSP). The proposed method increases the search diversity without requiring additional evaluation cost by introducing mutation whenever the ant colony encounters a local best. Twenty-two maps from the TSPLIB library were used to evaluate the effectiveness of this modified algorithm, which performed better than previous mutation-based ACO methods and traditional ACO. The results demonstrated the ability of the method to avoid local best bests and produce more optimal solutions while maintaining computational efficiency.

Ratanavilisagul et al., [5] In order to overcome the early stagnation of the local optimum, this study builds on previous work by introducing an improved ACO algorithm that combines binary heuristics and pheromone re-initialization. When ants were stuck, the algorithm used the re-initialized pheromones, which greatly enhanced the heuristic capabilities. The method outperformed previous versions of ACO, such as multi-colony techniques, in terms of solution quality when tested on twenty-three TSPLIB maps. It is a powerful choice for difficult optimization problems such as TSP because of the study's focus on finding a balance between computational feasibility and solution diversity.

Xu et al.,[6] This paper presents a hybrid optimization technique that addresses the Traveling Salesman Problem (TSP) by combining Firefly (FA) and Ant Colony Optimization (ACO). By optimizing the initial parameters of ACO, FA increases the convergence rate and reduces the probability of stagnation in the local optimum. According to experimental data, the hybrid algorithm outperforms traditional ACO techniques in terms of path optimization and computation time. The method is proven to be adaptable to complex optimization challenges by dynamically changing the problem size. This study demonstrates the utility of hybrid metaheuristics in enhancing the flexibility and effectiveness of ACO.

Mandal et al.,[7] Using the Traveling Salesman Problem (TSP) model, this work proposed an improved Ant Colony Optimization (ACO) algorithm for DNA sequencing tasks. To increase the answer quality and simplify the search process, the method used advanced pheromone updating algorithms and mutation procedures. Experimental results demonstrated the algorithm's ability to surpass traditional variations of the TSP in pathfinding accuracy and computational efficiency. By optimizing the unique constraints present in DNA sequencing, the technique demonstrated exceptional effectiveness in handling complex routing situations. This study demonstrated how the TSP can be used to solve highly specialized variations of the TSP with domain-specific modifications.

Fejzagićet al., [8] We investigated the use of parallel ACO to solve large TSP cases in an attempt to reduce computation time. Using the Task Parallel Library (TPL), the study found that parallel implementation improves the time efficiency of the algorithm while maintaining the quality of the answer. Since standard algorithms are too slow for large-scale TSP problems, the results demonstrate the suitability of parallel ACO. This study demonstrates the importance of using parallel computing to solve complex combinatorial optimization problems.

Zeng et al., [9] This study presented an improved ACO algorithm using dynamic heuristics to solve the Traveling Salesman Problem (TSP) using replenishment arcs. The model optimized the utilization of people and equipment in the transportation sector and included cumulative travel limits. The results validated the application of ACO to dynamic variations of TSP by showing better performance in determining the shortest paths under complex constraints.

Xu Li et al., [10]proposed the average absolute eigenvalue of the pheromone matrix (AAEPM) as a metric for assessing closeness. By analyzing the eigenvalues of the pheromone matrix, AAEPM provided a numerical evaluation of the convergence of the algorithm. The index demonstrated flexibility across different issue metrics and parameter configurations, providing a new viewpoint for tuning and tracking ACO performance.

Houssein et al., [11]Proposed a strategy to narrow the solution space in order to solve the multiple traveling salesman problem (MTSP). Efficiency was given top priority in this method with the distribution of vendors to cities and scheduling of their trips. Its greater performance in reducing trip expenses and processing time has been proven through comparative trials, demonstrating its usefulness for real-world uses including resource allocation and logistics.

Sheng et al., [12]proposed SOS-MMAS, a hybrid approach that optimizes solutions to the Traveling Salesman Problem (TSP) by combining the Max-Min Ant System (MMAS) with Search for Symbiotic Organisms (SOS). This technology optimizes important elements such as guiding weight and pheromone effect to increase proximity and flexibility.Experimental results show that SOS-MMAS outperforms SOS-ACO and standard ACO in terms of speed, flexibility, and solution quality, especially in large-scale TSP scenarios. With faster iterations and lower average errors, it has proven effective in solving real-world routing and scheduling problems. The study also suggested that SOS-MMAS could be extended to other optimization problems through the use of local research methodologies.

Wu et al., [13] Ant system (AS) and ant colony system (ACS) are two of the five classical ant colony optimization (ACO) algorithms modified in this work to answer the team routing problem in TOP. The modifications sought to maximize the total profits while distributing rewards along the routes. To optimize the solutions, an innovative ant team selection process and an iterative optimization process were implemented. The elite ant system (EAS) showed superior stability in minimizing the profit disparity between routes, while the ACS was the best at maximizing profits, according to the experimental results.

Yan et al., [14] The non-dominated multi-objective ant colony optimization (NSMOACO) algorithm is presented in this study to solve the path planning problem of unmanned underwater vehicles (UUV) in target search missions. The program uses tangent flight and adaptive mechanisms to dynamically adjust parameters in order to achieve two competing goals: search gain and flight duration. The global search capabilities of the algorithm are improved and premature convergence is avoided by combining non-dominated sorting and a novel pheromone updating technique. Comparative tests showed that NSMOACO performed better in terms of convergence speed and solution quality than other multi-objective optimization methods, including traditional ACO. This study highlighted the usefulness of NSMOACO in solving multi-objective problems in challenging real-world situations.

Prado et al., [15] Variations of Ant Colony Optimization (ACO) in dynamic optimization contexts were evaluated in this study, with a particular focus on vehicle routing issues under changing conditions. By introducing criteria to evaluate the adaptability of the algorithm, the study demonstrated how well ACO can adapt to changes in constraints and objectives in real time. The proposed changes increased the efficiency of decision-making in dynamic situations, allowing ACO to successfully deal with changing demands. Experimental results showed improved performance compared to existing methods in terms of computational speed and solution quality. For real-world applications such as supply chain and logistics optimization, this work underscores the importance of dynamic adaptation in ACO.

Tang et al., [16] This paper presents an Ant Colony System (ACS)-based approach to improve logistics scheduling, focusing on the Multi-Trip Seller Problem (MTSP) in hub cities. In order to minimize costs and fairly distribute routes among sellers, the proposed ACS-MTSP algorithm takes into account hub cities with different business needs. Experimental results demonstrate how well it can balance workload distribution and route duration.

Kothari et al. [17] A comprehensive analysis of heuristic algorithms, such as ACO, for large-scale TSP examples is performed in this paper. The results highlight the competitive performance of ACO in finding the balance between computation time and solution quality. The paper also emphasizes how hybrid algorithms, which combine ACO with methods such as particle swarm optimization and genetic algorithms, can be used to solve scaling and optimization problems in TSP.

Thongpiem et al., [18] This study proposed a hybrid approach to improve the quality of TSP solutions by integrating ACO and genetic algorithms. Genetic crossover and pheromone re-initialization were used to increase search diversity and prevent local optimal results. The hybrid algorithm consistently outperformed traditional ACO algorithms in tests on 23 TSPLIB datasets. This strategy confirmed the advantages of integrating heuristics for complex optimization problems. The study demonstrated the efficiency of the hybrid approach in producing better results.

Han et al. [19]The Color Mobile Salesman Problem (CTSP), a variant of TSP, was solved using an improved ACO. The study improved the algorithm's ability to find optimal solutions in large-scale CTSP situations by enhancing pheromone updating through the use of the ITÔ process. According to the experimental data, the improved ACO performed better than other algorithms in terms of computational speed and solution quality. Task allocation is a crucial component of real-world problems such as intelligent transportation systems and multi-task collaboration, where this approach is particularly useful.

Cheong et al., [20] The study evaluated the effectiveness of ACO using algorithms such as Kohonen and Christofides and investigated variations in ACO parameters for solving TSP. The study showed how parameter adjustments affected optimization results across different variables including pheromone levels, colony size, and evaporation rates. The results confirmed the strength of ACO as a heuristic approach and showed it to be competitive in small to medium-sized TSP situations.

De Oliveira et al., [21] In this paper, ant colony optimization (ACO) techniques are investigated for the traveling salesman problem (TSP) with dynamic demands. The P-ACO algorithm, which modifies the pheromone memory to solve dynamic problems, is presented. The importance of local search and parameter settings is highlighted by comparing P-ACO with the max-min ant system (MMAS). The results show that MMAS performs better when using local search, while P-ACO performs better in dynamic conditions without it. The study also emphasizes the importance of adaptive configurations to improve ACO algorithms in dynamic combinatorial optimization problems.

Dewantoro et al., [22] The hybrid ACO-TS technique was developed in this study by combining ant colony optimization (ACO) and taboo search (TS) to solve the TSP problem. By enhancing path optimization and accelerating convergence, the hybrid approach enhanced the performance of ACO. Experimental results showed that the ACO-TS algorithm outperforms the standalone ACO, especially when it comes to avoiding local optima and reducing computation time.

Duan et al., [23]This study presented a new approach to solving the TSP problem using a probe machine model, which accelerates problem solving using DNA-based computing techniques. When it comes to handling small-sized TSP portfolios, the PROBE4TSP solution outperforms traditional techniques with significant gains. The work showed how non-Turing computational models can be used to solve NP-hard problems such as TSP.

Ekmekciet al., [24]The study proposed a new form of ACO called Ant Colony Optimization Memorizing Better Solutions (ACO-MBS), which optimizes pheromone updates based on solution costs. ACO-MBS enhanced exploration and exploitation capabilities by including memory-based methods. Comparative research revealed that ACO-MBS performed better on standard TSP problems than regular ACO versions, obtaining higher convergence rates and higher quality solutions.

Fei et al., [25] ACO was extended to include multi-objective optimization with a focus on vehicle routing that balances fuel usage and trip duration. ACO and a dynamic approach were used to increase computational efficiency and solution quality. When compared to other heuristic algorithms, the method produced competitive results. Sheng (2022) also used ACO to address multi-objective vehicle routing problems with dynamic constraints, such as traffic conditions. In real-world logistics situations, the method demonstrated greater flexibility and adaptability.

Latha et al. [26] In this paper, the application of Ant Colony Optimization (ACO) to routing protocols for Traveling Salesman Problem (TSP) applications in Wireless Body Area Networks (WBAN) is investigated. A novel variation of ACO is proposed in combination with energy and distance-based TOPSIS to reduce packet transmission delays. The end-to-end delays are significantly reduced through the comparative study, indicating the potential of the algorithm in emergency health monitoring situations.

Liu et al.,[27]The mucus mold ant colony fusion algorithm (SMACFA) is presented, which improves the ant colony optimization (ACO) for TSP solutions. The combination of the mucus mold algorithm (SMA) and ACO in the model shortens the convergence time and prevents the algorithm from reaching the local best practices. When the experimental results are compared with the original ACO algorithm, the path length is improved by 1.42%, and the convergence time is reduced by 73.55%. In addition, compared with other optimization algorithms, the fusion performance is better. Large TSP cases benefit from the increased computational efficiency of this hybrid approach and the solution quality.

Meng et al., [28] The Generalized Travel Salesman Problem (GTSP) was investigated using a modified ACO technique. In order to enhance the path length optimization and maintain the job balance among multiple travel agents, this improved approach incorporates a binary choice algorithm. According to the study, the modified ACO outperforms the traditional techniques in terms of convergence and stability. The proposed method showed better results in a variety of conditions and was particularly successful in distributing tasks among agents in a balanced manner. For this reason, the method is useful for practical applications such as task scheduling and vehicle routing.

Murugananthan et al., [29]The main objective of this study was to improve the Ant Colony Optimization (ACO) algorithm to handle large-scale traveling retailer problems. The incorporation of genetic algorithms to balance exploration and exploitation, as well as adaptive pheromone updates, were important developments. These changes significantly increased the convergence rate of ACO and the solution quality. On benchmark datasets, experimental results showed that the algorithm outperformed traditional ACO algorithms in solving difficult optimization problems. For large datasets, the study highlighted the scalability and adaptability of the hybrid ACO approach.

Qian et al., [30]This study is designed as a kind of traveling salesman problem, and a multi-objective ant colony system (MOACS) is proposed to handle multi-agent pick-up and delivery tasks. The algorithm uses dual pheromone sets to simultaneously maximize competing objectives, such as task completion speed and workload balancing. The well-balanced exploration and exploitation approach and creative pheromone updating rules improve the flexibility of the algorithm. According to the experimental results, MOACS produces better solution quality than traditional ACO and other heuristic techniques. The study demonstrates how multi-objective optimization can be used to address challenging logistics problems in the real world.

Sharma et al., [31] This study aims to solve dynamic vehicle routing problems using real-time ACO. By using real-time traffic data to dynamically update pheromone levels, the software enables faster and more accurate routing decisions. The ACO is modified to account for time-dependent travel expenses in multi-objective vehicle routing, following a similar technique was introduced this authorFor sectors such as supply chain management and logistics that rely on real-time optimization, these developments are particularly useful. Both studies show that in dynamic contexts, real-time ACO can significantly increase operational efficiency.

Silalahi et al., [32] The Traveling Salesman Problem (TSP) was solved in this work using Ant Colony Optimization (ACO). Performance tests of the algorithm on a variety of paths showed that it could identify optimal and near-optimal solutions. The effectiveness of ACO was attributed to its powerful pheromone updating mechanism, which guided the ants towards efficient paths. The study demonstrated how well ACO can handle small to medium-sized datasets. This paper highlights the promise of ACO as a heuristic tool for generative optimization problems.

Steven et al., [33] By combining clustering methods with ACO, the study addressed the multi-traveler retailer problem (MMTSP). The MMTSP was partitioned into multi-traveler retailers in order to solve the problem efficiently using k-means and clustered clustering. The results showed that while clustered clustering with ACO gave better results than k-means, it also took longer to compute. The effectiveness of the approach in simplifying complex optimization tasks was confirmed by simulations performed on the TSPLIB dataset. This strategy confirmed the importance of clustering for improving ACO functions.

Stodola et al.,[34] Node clustering, adaptive pheromone evaporation, and novel termination conditions are the three innovative strategies used in the adaptive ACO algorithm in this study. By clustering nodes according to proximity, clustering increased the solution diversity and search efficiency. The termination condition depends on the population diversity, but adaptive pheromone evaporation exploited the information entropy to avoid stagnation. The approach outperforms state-of-the-art techniques in terms of convergence speed and solution quality when tested on 30 TSPLIB instances. These advances address some of the major drawbacks of the traditional ACO algorithm.

Sun et al., [35] The problem of multiple travel sellers with revisitable cities (MTSPR) was addressed in the study using a unique ACO algorithm. For revisitable cities, a balanced path selection approach was implemented, ensuring efficient path generation. To further improve the solution quality, the algorithm used a local binary search to optimize the elite ant path. When addressing the limitations of MTSPR, comparative tests showed that the proposed ACO performed better than alternative algorithms. This study demonstrated how ACO can be used for resource planning and logistics.

Thong-ia et al., [36] proposed the Gene-Ants algorithm, which overcomes the early-stage optimization limitations of ACO by combining genetic algorithm (GA) and ACO. Selectivity, exchange, and mutation are some of the genetic operations of the genetic algorithm that help avoid the local optimality problem that ACO usually faces. Tests on several TSP benchmarks have shown that the Gene-Ants algorithm performs better than the basic ACO algorithm in terms of convergence rate and global optimization. This hybridization makes it possible to provide a more reliable TSP solution.

Tuani et al., [37] An improved solution to TSP using a three-option local search in a heterogeneous adaptive ACO is presented. By continuously adjusting the parameters throughout the search process, the model enables the algorithm to successfully balance exploration and exploitation. Without the need for pre-defined parameters, the self-adaptive function of the algorithm improves performance and reduces the amount of time required for human adjustment. For large-scale TSP examples, experimental results show that the proposed approach outperforms traditional ACO methods.

Wang et al., [38] A better pheromone update model is introduced in ACO to address the multiple traveler-supplier (MTSP) problem with constraints such as capacity and time frame. The approach minimizes path length while meeting capacity and time requirements by solving the MTSP by combining a single tree with a minimum span ACO. The search efficiency and solution quality are improved by the hybrid approach. This development demonstrates the ability of ACO to solve increasingly complex variations in TSP under realistic logistical constraints.

Wang et al., [39] Modifying parameters such as α and β was proposed to enhance convergence in optimizing ACO-based TSP parameters. By introducing the hybrid symbiotic organism search (SOS) and ACO (SOS-ACO) technique, the study improved the quality of the result by adaptively optimizing the parameters. SOS-ACO was able to achieve solutions that were within 2.33% of the best TSP solutions, according to the results. Using a variety of TSP cases from TSPLIB, this author evaluated the model and demonstrated its effectiveness. This technique greatly simplifies the process of manually tuning ACO parameters.

Chen et al., [40]This paper presented an ant colony system (ACS)-based approach to improve logistics scheduling for the multiple traveling vendor problem (MTSP) with hub cities. In order to reduce the cost and fairly distribute the route among salespeople, the proposed ACS-MTSP algorithm takes into account hub cities with different business requirements. The results of the experiments showed how successful it was in shortening the path lengths and distributing the load evenly.

Chang et al., [41]K-means clustering was used in this work to improve the TSP solution efficiency of ACO. K-means was used to cluster cities into clusters, and before merging paths, ACO was applied independently to each cluster. In some city distributions, this method has improved performance while reducing compute costs by more than 30%. “Significant promise for improving ACO in complex TSP settings has been shown through this hybrid approach.”

Table 1- Related work summary table

#Reff	Author (Year)	Method	Dataset	Advantage	Disadvantage	Result	Accuracy
[1]	Scianna et al., (2024)	TSPLIB	Parallelization enhances performance on large problems	Still faces scalability issues for very large datasets	Parallel ACO provided better scalability than standard ACO	High efficiency	
[2]	Baydogmus et al., (2022)	Parallelized ACO	TSPLIB, 5 problems	Reduced memory usage, faster due to parallelization	Time complexity increases with number of colonies	50% faster than normal ACO operation	
[3]	Dou et al., (2024)	ACO for Multiple Traveling Salesmen with Constraints	VCMTSP benchmark set from TSPLIB	Effective for handling accessibility constraints	Requires further improvement for complex datasets	ACO and GA both addressed VCMTSP, but performance could be improved	
[4]	Ratanavilisagul et al., (2017)	Modified ACO with Pheromone Mutation	TSP (TSPLIB)	Avoids local optima, enhanced search diversity	Increased computational cost due to mutation steps	Outperformed standard ACO in solution quality	Better solutions than standard ACO
[5]	Ratanavilisagulet al., (2018)	Modified ACO with Leader and Re-initialization	TSP (TSPLIB)	Re-initialization prevents local optima trapping	Higher complexity with multiple colony re-initializations	Outperformed standard ACO and PACO-3OPT	Improved solution quality and convergence
[6]	Xu et al., (2023)	ACO and FA hybrid	TSPLIB standard	Improved convergence and local optima avoidance	Depending on the FA's initial performance	Enhanced path optimization and decreased processing time	High (increased accuracy in pathfinding)
[7]	Mandal et al., (2022)	Modified ACO for Generalized TSP	GTSP benchmark set	Good stability and optimization	Increased complexity with more agents	2.59% optimization in average path length	Optimized by 2.59% in average path

				accuracy		over ACO	
[8]	Fejzagić 2013	Parallel ACO	TSP (varied city sizes)	Improved execution time via parallelization, same solution quality	Increased complexity in parallelization implementation	Parallel ACO reduced execution time but with similar quality	Speed improvement, solution quality maintained
[9]	Zeng et al., (2021)	Enhanced ACO using dynamic heuristic data	TSP in transportation situations with replenishing arcs	Effective in resolving dynamic limitations	High processing demands for bigger datasets	shortest routes when cumulative travel restrictions are in place	For dynamic restrictions, accurate
[10]	Xu Li et al., (2024)	AAEPM for assessing convergence	ACO situations that were simulated	Strong convergence assessment independent of parameters	does not immediately increase the effectiveness of pathfinding	Precise ACO convergence monitoring	Precise assessment of convergence state
[11]	Houssein et al., (2024)	Space Reduction ACO for MTSP	MTSP with varying number of tasks and salesmen	Reduced solution space, faster computation time	Performance can drop with very large datasets	Outperformed classical methods in execution time and path length	Best execution time, competitive in path length
[12]	Sheng et al., (2022)	SOS-MMAS Hybrid ACO	TSPLIB	Improved task scheduling efficiency, avoids premature convergence	May increase computational complexity for larger datasets	SOS-MMAS outperforms standard ACO in TSP problem solving	High performance in large TSP instances
[13]	Wu et al., (2024)	ACS and EAS were modified for TOP.	Artificial TOP datasets	Improved route balance and profit maximization	restricted to some TOP variations	Enhanced efficiency and balanced earnings across routes	High (ACS excelled in profit maximization)

[14]	Yan et al., (2024)	For MOTSP, NSMOACO	MOTSP scenarios that were simulated for UUV	Superior adaptive algorithms and worldwide search capabilities	computationally demanding for extensive MOTSP	improved results and quicker convergence than with conventional techniques	High (Outperforming in jobs involving multi-objective optimization)
[15]	Prado et al., (2024)	Ant Colony Systems for Dynamic Vehicle Routing	Vehicle Routing Problem (VRP)	Adapts to real-time dynamic changes, good for real-time decision making	Requires fast computation for real-time changes	Evaluated multiple ACO variants for dynamic VRP, better real-time performance	95%
[16]	Tang et al., (2023)	Ant Colony Adaptive Optimization (AACO-LST)	45 TSP instances	faster convergence and more effective search	Large-scale dimensional catastrophe dilemma TSP	Comparing AACO-LST to ACS, the quality of the solution increased by 79%.	79%
[17]	Kothari et., (2024)	Meta-Heuristic Algorithms for TSP	256-city TSP dataset	Comprehensive comparison of multiple algorithms	Does not provide a clear comparison of hybrid approaches	Christofides was most cost-efficient, Simulated Annealing fastest	
[18]	Thongpiem et al., (2024)	Ant colony algorithm and hybrid genetic algorithm (HGAACO)	TSPLIB (23 instances)	Combining GA and ACO improved performance over MACO-LR.	Additional computational resources are needed for the hybrid technique.	HGAACO improved the speed and quality of the MACO-LR solution.	100%
[19]	Han et al., (2020)	Improved ACO for Large Scale	Large-scale CTSP problem	Optimized for large scale,	High computational cost,	Better performance than	Improved solution quality

		CTSP		avoids local optimum with ITÔ process	complexity in pheromone updating	classical algorithms for large-scale CTSP	
[20]	Cheong et al., (2017)	ACO with parameter variation	TSP (various datasets)	Better performance with varying colony sizes and other parameters	Requires fine-tuning for optimal results	ACO provided competitive results compared to other algorithms	Optimized route selection.
[21]	de Oliveira et al., (2021)	ACO for dynamic TSP with dynamic demands	TSP with dynamic demands	Enhanced performance in dynamic environments using P-ACO	P-ACO not as effective with local search components	P-ACO outperformed MAX-MIN Ant System (MMAS) without local search	Enhanced pheromone reuse.
[22]	Dewantoro et al., (2019)	Hybrid ACO-TS (ACO with Tabu Search)	TSP (standard problems)	Better route optimization and faster runtime	Hybrid algorithm increases complexity	ACO-TS outperformed standard ACO in route quality and runtime	Superior computational performance.
[23]	Duan et al., (2024)	Probe machine-based approach for TSP	TSP (various sizes)	Significant speedup compared to classical solvers for small-scale problems	Only effective for smaller problem sizes, not for large-scale problems	Faster than classical solvers for small TSP instances	High performance for smaller instances
[24]	Ekmekci et al., (2019)	ACO-MBS (Memorizing Better Solutions)	TSPLIB (eil51, kroA100)	Increased exploitation ability while maintaining exploration	Convergence speed can be slower under certain conditions	Outperformed standard ACS in convergence speed and solution quality	Achieved high accuracy in benchmark problems (eil51 and kroA100)
[25]	Fei et al., (2022)	Graph Convolutional	TSP datasets, engineerin	Improves initial convergen	Complex algorithm, requires	Outperformed other classical	High accuracy, faster

		Network Improved ACO (GCNIACO)	g application	ce speed, enhances local optimum escape	tuning for larger instances	algorithms in solution quality	convergence
[26]	Latha Ra et al., 2023	Energy-distance based TOPSIS-ACO	Wireless Body Area Networks (WBAN)	Improved end-to-end delay and packet routing delay management	High delay without routing strategy, delay under certain methods	Improved delay times compared to non-routing	0.126 ms (end-to-end delay)
[27]	Liu et al., (2020)	Slime Mold-Ant Colony Fusion Algorithm (SMACFA)	TSPLIB (chn31)	Enhanced global optimization, faster convergence, reduced complexity	Susceptible to local optimization in certain settings	1.42% improvement in path length over ACO, faster convergence	Improved by 1.42% in path length
[28]	Meng et al., (2019)	Modified ACO with 2-opt algorithm for GTSP	GTSP with 16 cities	Better optimization, task balancing among agents	Increased variance with more agents	2.59% average path length improvement over ACO	Optimized by 2.59% in average path length
[29]	Muruganant hanet al., (2023)	Ant Colony Optimization (ACO)	TSPLIB and customized datasets	Flexibility and ability to adapt to dynamic scenarios	Susceptible to slow convergence for large datasets	Outperformed Cplex optimizer in multiple test cases	Outperformed Cplex optimizer
[30]	Qian et al., (2024)	Multiobjective Ant Colony System (MOACS)	Pickup and Delivery tasks	Optimizes multiple objectives (working time and workload balance)	Complexity increases with more agents and tasks	MOACS outperforms other ACS-based multi-objective algorithms	High performance in multi-agent scenarios
[31]	Sharma et al., (2024)	ACO-based Energy Efficiency Optimization for IoT-Cloud	IoT and Cloud Computing Resources	Reduces energy consumption in cloud environments	High adaptability required for real-time IoT environments	Outperformed conventional resource allocation strategies	Significant energy savings and reduced operational cost
[32]	Silalahiet al., (2019)	ACO for TSP	TSPLIB	Faster execution compared to exact methods	Struggles with larger cases	ACO was faster than exact methods in solving TSP	Execution time significantly reduced

[33]	Steven et al., (2017)	Clustered ACO for MMTSP	TSPLIB	Clustering improves route optimization	Agglomerative clustering takes longer than K-means clustering	Agglomerative ACO outperforms K-means clustering and standalone ACO	Best route with agglomerative ACO
[34]	Stodola et al., (2022)	Adaptive ACO with Node Clustering	TSPLIB (51-2392 nodes)	Improved performance, reduces execution time and local optimum risk	Parameter settings still affect performance	Outperformed other ACO-based methods on benchmark tests	Higher convergence speed, better solutions
[35]	Sun et al., (2024)	RACO (ACO for MTSPR)	TSPLIB	Effective path design that balances salesmen's paths	Construction of complex paths necessitates careful task balancing.	When solving MTSPR, RACO performs better than other algorithm.	Better results with well-balanced routes.
[36]	Thong-iaet al., (2023)	Ants Gene (ACO with GA)	TSPLIB	avoids local optima and improves global search by combining ACO and GA.	longer computation times as a result of the hybrid technique	Gene-Ants fared better in global optimal solution discovery than simple ACO.	Enhanced rate of convergence and quality of the solution
[37]	Tuaniet al., (2020)	Adaptive Heterogeneous ACO with 3-opt Local Lookup	TSPLIB	Parameters are adaptively adjusted to prevent premature convergence.	costly to compute with large-scale TSP	Heterogeneous ACO performed faster and better than traditional algorithms.	Increased resilience and faster convergence
[38]	Wang et al., (2020)	ACO for MTSP with an	MTSP with time window	A better pheromone model	More intricacy as a result of	superior search effectiven	Shorter routes and improved

		Enhanced Pheromone Model	and capacity restrictions	that manages capacity and logistics issues	the hybrid pheromone model	ess and solution quality compared to alternative algorithms	logistics optimization performance
[39]	Wang et al., (2021)	ACO and SOS hybrid	TSPLIB standard	Enhanced global search capability and convergence rates	intricate tweaking of parameters	Solutions that fall between 2.33% and the established optimal values	High (deviation of 2.33% from ideal)
[40]	Chen et al., (2024)	ACS-MTSP (Multiple TSP with hub cities)	TSPLIB (att48, kroA100, etc.)	Effective in reducing traveling salesmen costs	Still computationally complex for larger datasets	Path lengths minimized with stable results	High performance
[41]	Chang et al., (2017)	ACO combined with K-means clustering	TSP (various city distributions)	Reduced computational cost, improved performance for specific distributions	May not generalize well to all city distributions	32% faster than unclustered ACO	Improved in specific setups

Discussion

The examined literature shows that ant colony optimization (ACO) techniques for solving the traveling salesman problem (TSP) and its many variations have advanced significantly. Hybridizing ACO with other optimization techniques to improve performance is a popular trend.For example, [6] used Firefly algorithm (FA) for parameter optimization, which enhanced the convergence speed and flexibility, while [7] Integrating mutagenesis techniques into ACO for DNA sequencing. Similarly, [12] improved task scheduling efficiency using a hybrid ACO for Multi-Depot Multiple TSP (MMTSP). The effectiveness of hybrid techniques in solving limited and multi-travel seller problems has been demonstrated by [3] and [11], demonstrating the flexibility of ACOs in dealing with difficult situations. By optimizing the competing objectives of unmanned underwater vehicle trajectories, Yan et al. (2024) developed a nondominated sorting multi-objective ACO (NSMOACO) which further developed this idea.

Studies such as [2] and [8]have shown how parallel ACO shortens execution times without sacrificing solution quality, making parallel computing another important issue. [13] and [9] have demonstrated how these advances have improved ACO's suitability for large-scale applications including supply chain optimization and logistics. In the literature, adaptive mechanisms are also frequently discussed. For example, [10]introduced the AAEPM for convergence evaluation, which offers reliable monitoring without directly improving pathfinding. Adaptive pheromone updates by[34] and dynamic heuristic algorithms by [27] also demonstrate the ability of ACO to manage dynamic and large-scale optimization problems. Even with these developments, there are still limitations. As shown by [7]and [6], because new technologies require significant resources, hybrid approaches often increase computational complexity. [8] point out that although parallel ACO increases scalability, its implementation may be hardware dependent. [4],[5] and [39] note that tuning parameters is still a great difficulty and that performance mostly depends on manual changes. While new approaches such as enhanced mutation techniques [19] and [10] address convergence, they do not immediately increase computing efficiency. According to [17], addressing these issues requires studying lightweight hybrid models in order to balance efficiency and solution quality. Studies such as [35] and [30] suggest that incorporating machine learning methods can automate parameter tuning and adjust ACO to real-time conditions. In addition, as suggested by potential works such as [15] and [16], using advances in quantum computing may create new opportunities to scale ACOs. Significant advances in hybridization, parallelism, and adaptability have been highlighted in the literature, confirming ACO's position as a reliable tool for solving challenging optimization problems in a variety of domains.

Challenges and Future Directions

ACO has made progress, but there are a number of barriers to its wider use. Although effective, hybrid models come with a computational cost and require advanced technology for parallel processing [8],[7]. Furthermore, parameter adjustment is critical to the effectiveness of an ACO, and less-than-optimal settings often lead to suboptimal results. While convergence evaluation is addressed by frameworks such as AAEPM [10], parameter selection is not automated. Future studies should investigate frameworks that rely on machine learning to adjust parameters to automate optimization, as well as lightweight hybrid models that strike a balance between computational efficiency and solution quality [15], [16]. ACO may be able to handle previously unheard-of problem sizes through the use of quantum computing [17].

Conclusion

The traveling salesman problem (TSP) is the mainstay of combinatorial optimization, and is known for its complexity and wide range of practical applications in resource management, communications, and logistics. Inspired by the feeding habits of ants, ant colony optimization (ACO) has become a reliable and flexible way to address TSP. The program can efficiently find optimal or near-optimal solutions thanks to its iterative method, which relies on pheromone trails and heuristic information. Significant developments over time have improved ACO capabilities. By solving problems such as local optimal slack and scalability problems, hybrid models - which mix ACO with free algorithms - have enhanced their performance. These integrations have proven to produce high-quality solutions and faster convergence rates, especially in dynamic and multi-objective contexts. Moreover, the scalability of ACO has been transformed by parallel computing frameworks, which allow efficient processing of large data sets and solving scheduling and logistics problems at industrial scale. In dynamic routing applications, when variables such as traffic, time constraints, and resource availability change in real time, ACO has also shown remarkable adaptability. Their ability to adapt to these modifications shows how versatile they are at solving difficult problems in the real world. ACO has proven to be a useful and reliable solution to dynamic optimization problems, from emergency response and underwater navigation to vehicle routing in logistics. Even with its improvements, ACO still suffers from drawbacks, such as computational overhead in parallel and hybrid models, the need for human parameter adjustment, and scalability in contexts with limited resources. In order to improve scalability and efficiency, future research should focus on creating lightweight hybrid frameworks, including machine learning for automated parameter adjustments, and investigating quantum computing. The revolutionary importance that the ACO plays in the TSP solution and its changes is highlighted in this review. This article emphasizes the continuing importance and potential of ACOs by addressing current limitations and synthesizing important improvements. ACO is positioned to continue to be a pillar of optimization research as science and technology advance, solving more complex problems in a variety of fields.

References

[1] M. Scianna, “The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems,” *Math. Comput. Simul.*, vol. 218, no. November 2023, pp. 357–382, 2024, doi: 10.1016/j.matcom.2023.12.003.

[2] G. K. Baydogmus, “A Parallelization Based Ant Colony Optimization for Travelling Salesman Problem,” *2022 1st Int. Conf. Inf. Syst. Inf. Technol. ICISIT 2022*, pp. 166–169, 2022, doi: 10.1109/ICISIT54091.2022.9873092.

[3] X. Dou, Q. Yang, X. Gao, and Z. Lu, “Benchmark for Multiple Traveling Salesmen Problem with Visiting Constraints,” *2024 11th Int. Conf. Mach. Intell. Theory Appl.*, no. 3, pp. 1–8, 2024, doi: 10.1109/MiTA60795.2024.10751704.

[4] C. Ratanavilisagul, “Modified Ant Colony Optimization with pheromone mutation for travelling salesman problem,” *ECTI-CON 2017 - 2017 14th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol.*, pp. 411–414, 2017, doi: 10.1109/ECTICon.2017.8096261.

[5] C. Ratanavilisagul and B. Pasaya, “Modified ant colony optimization with updating pheromone by leader and re-initialization pheromone for travelling salesman problem,” *ICEAST 2018 - 4th Int. Conf. Eng. Appl. Sci. Technol. Explor. Innov. Solut. Smart Soc.*, pp. 1–4, 2018, doi: 10.1109/ICEAST.2018.8434500.

[6] Y. Xu and X. Yao, “A Hybrid Approach Based on Ant Colony and Firefly Algorithm for Traveling Salesman Problem,” *2023 8th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2023*, pp. 167–171, 2023, doi: 10.1109/ICCCBDA56900.2023.10154838.

[7] A. K. Mandal and P. Kumar Deva Sarma, “Novel Applications of Ant Colony Optimization with the Traveling Salesman Problem in DNA Sequence Optimization,” *Proc. - 2022 IEEE 2nd Int. Symp. Sustain. Energy, Signal Process. Cyber Secur. iSSSC 2022*, 2022, doi: 10.1109/iSSSC56467.2022.10051206.

[8] E. Fejzagic and A. Oputic, “Performance comparison of sequential and parallel execution of the Ant Colony Optimization algorithm for solving the traveling salesman problem,” *2013 36th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2013 -*

Proc., pp. 1301–1305, 2013.

[9] X. Zeng, Q. Song, S. Yao, Z. Tian, and Q. Liu, “Traveling Salesman Problems with Replenishment Arcs and Improved Ant Colony Algorithms,” *IEEE Access*, vol. 9, pp. 101042–101051, 2021, doi: 10.1109/ACCESS.2021.3093295.

[10] X. Li, S. Member, L. Shi, C. Author, and J. L. C. Author, “Pheromone Matrix Eigenvalue-based Convergence Assessment Indicator for Ant Colony Optimization,” 2024, doi: 10.1109/MiTA60795.2024.10751702.

[11] F. A. Houssein, V. F. Kostyukov, and I. D. Evdokimov, “A method for solving the multi-traveling salesman problem based on reducing the size of the solution space,” *10th 2024 Int. Conf. Control. Decis. Inf. Technol. CoDIT 2024*, pp. 1729–1733, 2024, doi: 10.1109/CoDIT62066.2024.10708116.

[12] Y. Sheng, “A computational optimization research on ant colony optimization for the traveling salesman problem,” *J. Phys. Conf. Ser.*, vol. 2258, no. 1, 2022, doi: 10.1088/1742-6596/2258/1/012006.

[13] D. M. Wu, Q. Yang, C. J. Zhou, Z. L. Zheng, and J. Zhang, “Adapted Ant Colony Optimization for Large-Scale Orienteering Problem,” *GECCO 2024 Companion - Proc. 2024 Genet. Evol. Comput. Conf. Companion*, pp. 223–226, 2024, doi: 10.1145/3638530.3654270.

[14] Z. Yan, S. Cai, and S. Hou, “Non-Dominated Sorting Multi-Objective Ant Colony Optimization Algorithm for Route Planning of UUV Target Search,” *Chinese Control Conf. CCC*, vol. 3, pp. 4404–4409, 2024, doi: 10.23919/CCC63176.2024.10662714.

[15] L. M. Prado, T. Grandi, and C. C. Pietroboni, “Metrics for the Evaluation of Optimization Methods in Dynamic Problems: Application to Vehicle Routing Using Ant Systems,” *2024 7th IEEE Bienn. Congr. Argentina, ARGENCON 2024*, 2024, doi: 10.1109/ARGENCON62399.2024.10735838.

[16] K. Tang, X. F. Wei, Y. H. Jiang, Z. W. Chen, and L. Yang, “An Adaptive Ant Colony Optimization for Solving Large-Scale Traveling Salesman Problem,” *Mathematics*, vol. 11, no. 21, 2023, doi: 10.3390/math11214439.

[17] A. Kothari, V. R. N. S. Nikhil, N. Rohit, and A. Sahay, “Harnessing Meta-Heuristic Algorithms to Optimize Travelling Salesman Problem,” *2024 15th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2024*, pp. 1–9, 2024, doi: 10.1109/ICCCNT61001.2024.10726138.

[18] J. Thongpiem, N. Punkong, C. Ratanavilisagul, and S. Kosolsombat, “Hybrid Genetic Algorithm and Ant Colony Algorithm for Solving Travelling Salesman Problem,” *2024 IEEE 9th Int. Conf. Comput. Intell. Appl. ICCIA 2024*, pp. 69–73, 2024, doi: 10.1109/ICCIA62557.2024.10719116.

[19] S. Han, M. Xu, Q. Lin, Q. Li, and Q. Guo, “An Improved Ant Colony Optimization for Large Scale Colored Traveling Salesman Problem,” *Proc. - 2020 Int. Conf. Intell. Comput. Autom. Syst. ICICAS 2020*, pp. 400–405, 2020, doi: 10.1109/ICICAS51530.2020.00090.

[20] P. Y. Cheong, D. Aggarwal, T. Hanne, and R. Dornberger, “Variation of ant colony optimization parameters for solving the travelling salesman problem,” *IEEE 4th Int. Conf. Soft Comput. Mach. Intell. ISCMI 2017*, vol. 2018-Janua, pp. 60–65, 2017, doi: 10.1109/ISCMI.2017.8279598.

[21] S. M. de Oliveira, L. C. T. Bezerra, T. Stützle, M. Dorigo, E. F. Wanner, and S. R. de

Souza, “A computational study on ant colony optimization for the traveling salesman problem with dynamic demands,” *Comput. Oper. Res.*, vol. 135, p. 105359, 2021, doi: 10.1016/j.cor.2021.105359.

[22] R. W. Dewantoro, P. Sihombing, and Sutarman, “The Combination of Ant Colony Optimization (ACO) and Tabu Search (TS) Algorithm to Solve the Traveling Salesman Problem (TSP),” *2019 3rd Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM 2019 - Proc.*, pp. 160–164, 2019, doi: 10.1109/ELTICOM47379.2019.8943832.

[23] C. Duan, J. Liu, J. Xu, D. An, and Y. Bao, “A Novel Approach for Traveling Salesman Problem Via Probe Machine,” *IEEE Int. Conf. Softw. Qual. Reliab. Secur. QRS*, pp. 713–723, 2024, doi: 10.1109/QRS62785.2024.00076.

[24] D. Ekmekci, “An Ant Colony Optimization Memorizing Better Solutions (ACO-MBS) for Traveling Salesman Problem,” *3rd Int. Symp. Multidiscip. Stud. Innov. Technol. ISMSIT 2019 - Proc.*, pp. 1–5, 2019, doi: 10.1109/ISMSIT.2019.8932768.

[25] T. Fei, X. Wu, L. Zhang, Y. Zhang, and L. Chen, “Research on improved ant colony optimization for traveling salesman problem,” *Math. Biosci. Eng.*, vol. 19, no. 8, pp. 8152–8186, 2022, doi: 10.3934/mbe.2022381.

[26] R. Latha, R. S. Kumar, B. S. Kumar, S. Rajalingam, and N. D. Tamizhnesan, “Routing Protocol using Ant Colony Optimization-Traveling Salesman Problem,” *Procedia Comput. Sci.*, vol. 230, no. 2023, pp. 515–521, 2023, doi: 10.1016/j.procs.2023.12.107.

[27] M. Liu *et al.*, “A Slime Mold-Ant Colony Fusion Algorithm for Solving Traveling Salesman Problem,” *IEEE Access*, vol. 8, pp. 202508–202521, 2020, doi: 10.1109/ACCESS.2020.3035584.

[28] L. Meng, Y. Lin, S. Qing, and F. Wenjing, “Research on Generalized Traveling Salesman Problem based on Modified Ant Colony Optimization,” *Proc. 31st Chinese Control Decis. Conf. CCDC 2019*, pp. 4570–4574, 2019, doi: 10.1109/CCDC.2019.8833167.

[29] V. Murugananthan, M. Y. E. S. Rehan, R. Srinivasan, M. Kavitha, and R. Kavitha, “Traveling Salesman Problem with Ant Colony Optimization,” *Proc. 2nd Int. Conf. Edge Comput. Appl. ICECAA 2023*, no. Icecaa, pp. 481–485, 2023, doi: 10.1109/ICECAA58104.2023.10212262.

[30] T. Qian, X. Liu, and Z. Zhan, “A Multiobjective Ant Colony System for Multi-Agent Pickup and Delivery,” *2024 11th Int. Conf. Mach. Intell. Theory Appl.*, pp. 1–8, doi: 10.1109/MiTA60795.2024.10751680.

[31] D. Sharma and A. S. Parihar, “Ant Colony Optimization-Based Energy Efficiency Optimization in IoT-Enabled Cloud Computing Resource Allocation,” *2024 Int. Conf. Adv. Comput. Res. Sci. Eng. Technol. ACROSET 2024*, pp. 1–8, 2024, doi: 10.1109/ACROSET62108.2024.10743955.

[32] B. P. Silalahi, N. Fathiah, and P. T. Supriyo, “Use of Ant Colony Optimization Algorithm for Determining Traveling Salesman Problem Routes,” *J. Mat. “MANTIK,”* vol. 5, no. 2, pp. 100–111, 2019, doi: 10.15642/mantik.2019.5.2.100-111.

[33] A. Steven, G. F. Hertono, and B. D. Handari, “Implementation of clustered ant colony optimization in solving fixed destination multiple depot multiple traveling salesman problem,” *Proc. - 2017 1st Int. Conf. Informatics Comput. Sci. ICICoS 2017*, vol. 2018-Janua, pp. 137–140, 2017, doi: 10.1109/ICICOS.2017.8276351.

[34] P. Stodola, P. Otřísal, and K. Hasilová, “Adaptive Ant Colony Optimization with node clustering applied to the Travelling Salesman Problem,” *Swarm Evol. Comput.*, vol. 70,

no. January, 2022, doi: 10.1016/j.swevo.2022.101056.

[35] B. Sun, W. Yu, Z. Wang, and J. Li, “Ant Colony Optimization for Multiple Traveling Salesmen Problem with Revisitable Cities,” *2024 11th Int. Conf. Mach. Intell. Theory Appl.*, pp. 1–6, 2024, doi: 10.1109/MiTA60795.2024.10751693.

[36] S. Thong-Ia and P. Champrasert, “Gene-Ants: Ant Colony Optimization with Genetic Algorithm for Traveling Salesman Problem Solving,” *2023 Int. Tech. Conf. Circuits/Systems, Comput. Commun. ITC-CSCC 2023*, pp. 1–5, 2023, doi: 10.1109/ITC-CSCC58803.2023.10212945.

[37] A. F. Tuani, E. Keedwell, and M. Collett, “Heterogenous Adaptive Ant Colony Optimization with 3-opt local search for the Travelling Salesman Problem,” *Appl. Soft Comput.*, vol. 97, no. xxxx, p. 106720, 2020, doi: 10.1016/j.asoc.2020.106720.

[38] M. Wang, T. Ma, G. Li, X. Zhai, and S. Qiao, “Ant Colony Optimization with an Improved Pheromone Model for Solving MTSP with Capacity and Time Window Constraint,” *IEEE Access*, vol. 8, pp. 106872–106879, 2020, doi: 10.1109/ACCESS.2020.3000501.

[39] Y. Wang and Z. Han, “Ant colony optimization for traveling salesman problem based on parameters optimization,” *Appl. Soft Comput.*, vol. 107, p. 107439, 2021, doi: 10.1016/j.asoc.2021.107439.

[40] Y. Chen and F. Tang, “An Ant Colony System Algorithm for Multiple Traveling Salesman Problem with Hub Cities,” *2024 11th Int. Conf. Mach. Intell. Theory Appl.*, pp. 1–7, 2024, doi: 10.1109/MiTA60795.2024.10751731.

[41] Y. C. Chang, “Using k-means clustering to improve the efficiency of ant colony optimization for the traveling salesman problem,” *2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017*, vol. 2017-Janua, pp. 379–384, 2017, doi: 10.1109/SMC.2017.8122633.