مجلة
كلية التــراث الجامعة

مجلة علمية محكمة
متعددة التخصصات نصف سنوية
العدد الحادي والأربعون
30 نيسان 2025
ISSN 2074-5621

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (4/7 /2014)

An Improved Ant Colony Optimization Approach Using Metaheuristics and Divide-and-Conquer for Solving the Traveling Salesman Problem

Zainab.a.Yakoob

University of Technology, College of Computer Science

Abstract

The Travelling Salesman Problem (TSP) is a Classic nondeterministic problem seeks to identify the shortest path which passes through every city in a single trip before arriving back at the starting location. Ant Colony Optimization (ACO) technique provides good results for TSP, but requires significant computational time. In this paper, some steps are presented to resolve the TSP problem. The description of the new suggested method is the partitioning of a problem into a number of subproblems that can be dealt with separately and combinedly solves them for more than once using ACO for each cluster. In this way, the amount of time required for computation is reduced. The experimental results point that the suggested algorithm provides significantly better results than ACO in the aspects of time required for computation and length of the determined optimal path.

Keywords: Traveling Salesman Problem, Ant Colony Optimization, Metaheuristic Algorithms, Divide-and-Conquer Strategy, Pheromone Update

1.Introduction

With the rapid expansion of current business environment and economy, the importance of path planning problems has grown in many areas. Effective path planning is essential for improving efficiency, lowering costs, and conserving energy in several industries such as logistics, traffic management, planning for cities, and robotics [1] [2]. Nature is full with different species, each of which lives in its own unique way. Natural selection is superior than human decision-making abilities. Humans have developed tools and strategies that simulate the behavior and structure of natural kinds for their benefit [49], [50], [51]. The computer transformed human computing and simulation capabilities [3]. The Travelling Salesman Problem (TSP) is one of the most well-known and extensively researched NP-hard problems that means the number potential tours increases according to the number of cities (n!) [4], [5], [6]. Checking all potential trips to find the best one for large (n) [4] [5]. The most essential generalization of the traveling salesman problem for transportation and logistics is that several vehicles with limited capacity must serve clients by visiting them during the prescribed time intervals [7], [8], [9]. For more than four decades, the traveling salesman problem has remained an entirely mathematical challenge[6]. The concepts of local search were expanded into metaheuristics, which are generic approaches for creating algorithms that may be applied to nearly any discrete optimization issue [10], [11], [12]. Metaheuristics are iterative processes with asymptotic convergence to global optimums. Metaheuristics, unlike estimation algorithms, are not dependent on the problem's characteristics. Metaheuristics include Genetic Algorithms, Gray Wolf Optimization, Evolutionary Computation, Cuckoo Search, Tabu Search, Artificial Bee Colony, Ant Colony Optimization (ACO), and others [7]. ACO approaches have a distinct approach, which is centered on gathering statistical data on those that produce successful solutions. This information is used in probabilistic greedy algorithms to predict what elements of the outcomes (graph ribs) have historically resulted in a minimal mistake. This group's methods are often based on wildlife analogies. The concept of ACO is an effort to mimic the behaviors of ants, which have very no vision and rely on smells left by their predecessors. A molecule with a strong odor, known as a pheromone, indicates precursor activity. The search history is accumulated, similar to the path to the colony of ants[8] [9].

This paper looks for to improve the ACO algorithm for TSP, by improving the pheromone update rules by setting upper and lower values for the pheromone limits, incorporating adaptive strategies by splitting ants into subgroups with various strategies to enhance distinction, improving the initial generation for the pheromone distribution, and finally fragmenting up the large TSP cities into smaller ones and processing each one separately to obtain the solutions, then recombining these solutions to obtain the smallest path ; depending on the principle of Divide and Conquer. The paper's remaining sections are organized as follows: The TSP is reviewed in Section 2, the ACO is presented in Section 3, the related work is shown in Section 4, the proposed method is displayed in Section 5, the evaluation and discussion are discussed in Section 6, and the study is concluded in Section 7.

2.Traveling Salesman Problem(TSP):

The Traveling Salesman Problem (TSP) is one of the most important problems in computer science and applied mathematics. The challenge is to find the shortest path that passes through all cities only once and then returns to the starting point [10]. TSP has problem characteristics:

- Input: A set of n cities, along with the distances between each pair of cities.
- Output: The optimal tour or sequence of cities that minimizes the total travel distance.
- Objective: To minimize the total length of the tour while visiting each city once.
- Complexity: The problem is NP-Hard, implying that no known algorithm can solve all instances optimally in polynomial time as the number of cities increases [11].

The TSP is one of the hardest problems in computer science, specifically in computational complexity theory. Its difficulties depend on the number of cities n increases; the number of possible paths grows factorially (n!), making brute force search impractical and necessitating the use of heuristic and metaheuristic algorithms [12]. The reasons why it is so difficult to solve are many; it Belongs to the NP-hard category where there is no polynomial time algorithm that can find the optimal solution for all cases and If it could be solve TSP efficiently, it will be able to solve all problems in the NP category quickly, which has not been proven yet [13] [14]. The number of solutions grows exponentially. If we have n cities, the number of possible arrangements of paths equals (n-1)! / 2 For example, if you have 10 cities, there are 181,440 possible routes! Which making Brute force impractical as the number of cities increases [15] . Finding the best answer is challenging due to computational complexity; as the number of cities increases, computers require a significant amount of time to solve the problem directly. Even the most exact methods, such Branch and Bound and Dynamic Programming, suffer as the size expands [16] [17]. This has resulted in an extensive adoption of heuristic and metaheuristic techniques, which yield near-optimal answers in an appropriate period of time for enormous datasets.

3. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a bio-inspired artificial intelligence algorithm developed by observing how real ant colonies search for food in nature. Ants communicate indirectly by depositing pheromone trails along the paths they travel, allowing other ants to follow these trails and gradually reinforce the shortest routes between their nest and the food source[18] [19]. As part of his doctoral thesis, Marco Dorigo first presented this approach in 1992. It has since been used to tackle combinatorial optimization issues, including the (TSP), which is the most well-known of these [20]. In recent applications, ACO has showed substantial utility, notably in areas such as maze generation and route planning, demonstrating both its benefits and potential limitations through varied experimental results (Napoli et al.). Understanding ACO's basic concepts helps improve its application to a wide range of scientific and industrial difficulties, confirming its significance within current optimization discourse [21] [22]. ACO is a methodology for developing metaheuristic strategies for problems relating to combinatorial optimization. Metaheuristic algorithms use two types of heuristics to avoid local optima: a constructive heuristic that starts with a null solution and adds elements to create a good solution, and a local search heuristic that starts with a complete solution and iteratively modifies some of its elements to improve it. The metaheuristic component enables the low-level heuristic to get solutions that are superior than those it could have obtained alone, even if iteratively [23][13]. Similar to other metaheuristics such as simulated annealing [24], tabu search [25], evolution strategies [26] and genetic [27], or bionomic [28] algorithms, ACO includes mechanisms to guide the exploration of the solution space and avoid premature convergence.

3.1 ACO Algorithm Steps

The operation of ACO involves several structured steps. Initially, a number of artificial ants are created and placed on random starting nodes. Several key parameters must be defined:

(τ_{ij}):Pheromone value on the path between city i and city j.
(η_{ij}): Heuristic value, typically $1/L_{ij}$, where L_{ij} is the distance between cities i and j.
(α):Weight of pheromone influence
(β):Weight of heuristic influence
(ρ):Pheromone evaporation rate
(Q): Constant used to control the amount of pheromone deposited [29].

The probability $P(i,j)$ that an ant located at city i will choose city j as its next destination is given by the formula:

$$P(i,j) = ([\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta})/(\sum_{k \in N(i)}[\tau_{ik}(t)]^{\alpha} \, [\eta_{ik}(t)]^{\beta}) \quad (1)$$

Where: P(I,j) is the probability between node i and j, τ_{ij} pheromones on the path between node i and j, k represent the ant number, $N(i)$ represents the numbers of city attached to the present city and possible to be visited by the ant

$$, \eta ij = 1/L_{ij} \quad (2)$$

Where L_{ij} is the length between node i and j.

After all the ants have passed and moved for the first time, the amount of pheromones that were placed on the current path by the ants will be updated, the pheromone value modified by the following equation:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}\,(t) + \rho\Delta\tau_{ij}\,(t) \quad (3)$$

Where ρ represents the amount of pheromone emitted from the path between city i and city j during the current ant iteration, and $\Delta\tau_{ij}$ represents the amount of increase in the pheromone set by the ants during the current iteration, as in the following equations:

$$\Delta\tau_{ij}\,(t) = \; Q/L^k, (i, j \in \, L^k\,) \quad (4)$$

Where Q represents the amount of pheromone density and L^k represents the total distance that the ant K has spent in the current cycle [30].

3.2 Max-Min Ant System (MMAS):

The Max-Min Ant System (MMAS) is one of the most effective ways to improve the standard ACO. This variation was created to address ACO's tendency to converge early to poor pathways. MMAS introduces two fundamental enhancements; firstly, Pheromone Limits: Set upper and lower boundaries on pheromone levels. (τ_{min} and τ_{max}). This ensures that not one route becomes excessively dominant early on, ensuring a balance between exploration and exploitation. Secondly, Selective Updating: MMAS Pheromone trails are updated by selecting the most successful ant, which may be this iteration or the most powerful so far. This suited motivation allows simpler for the method to provide high-quality results. MMAS employs these tactics to reduce the likelihood of stagnation while increasing the convergence date of the algorithm behavior. It has been particularly successful in addressing large TSP scenarios, making it a widely utilized foundation for more complicated ACO versions [31].

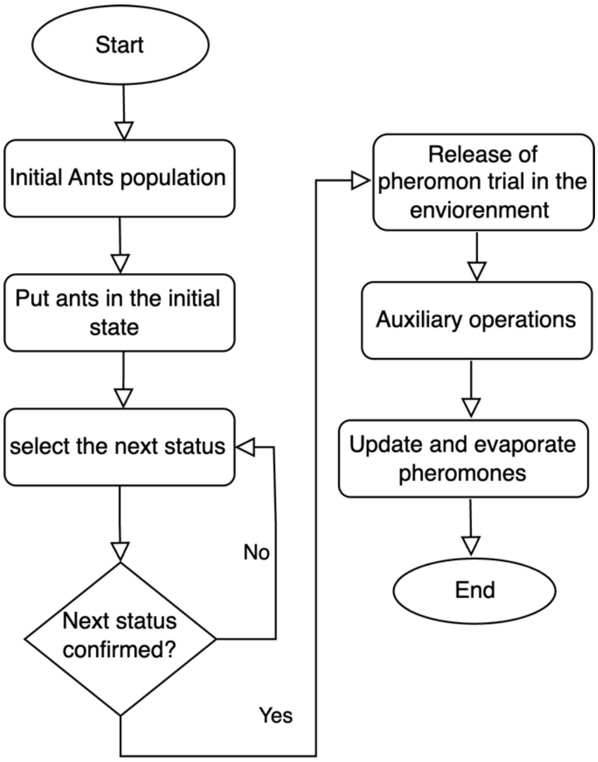


Fig. 1. Flowchart of Ant colony algorithm [31]

4. Related Work

The existing research on improved Ant Colony Optimization (ACO) algorithms for the Traveling Salesman Problem (TSP) is reviewed in this section. It summarizes and analyzes key approaches, results, and limitations in the literature. This section forms the foundation for identifying the knowledge gap and situating this study within the broader scholarly context. In [32], the Affinity Propagation Clustering algorithm is employed. Similar to k-means in terms of problem decomposition, it eliminates the need to predefine the number of clusters, thereby improving computational efficiency. Experimental results demonstrate that the proposed algorithm outperforms ACO in terms of both computation time and optimal path length. In [33] , a comparative analysis of the standard ACO and a parallelized version is conducted for solving the TSP. The study highlights that as problem size increases, execution time rises proportionately, and the runtime of ACO is significantly higher than that of exact algorithms. In [34], a hybrid algorithm named SMACFA is introduced, which combines the Slime Mold Algorithm (SMA) with ACO. SMA identifies high-quality path segments based on flow, distance, and conductivity, which are then used as fixed-point pairs within the ACO framework. This integration enhances convergence speed, solution quality, and reduces time complexity. Experiments using TSPLIB datasets confirm that SMACFA outperforms standard ACO and its variants, with improvements in robustness and reduction in randomness. In [35] , the DEACO algorithm is proposed, which modifies ACO parameters, particularly the pheromone evaporation rate. It utilizes clustering to determine the initial node for shortest path computation. Experimental findings reveal that DEACO achieves faster convergence and improved accuracy compared to traditional ACO. In [36], a variant named Focused ACO (FACO) is presented. FACO introduces a control mechanism to limit the deviation between new and previous solutions, thereby promoting a more focused search. Applied on the TSP Art Instances dataset, this method allows for local search integration and high-quality solutions for instances with up to 200,000 nodes within less than one hour of octa-core CPU time. In [37] the Graph Convolutional Network Improved Ant Colony Optimization (GCNIACO) is introduced. It uses a graph convolutional network to initialize pheromone levels, enhancing the guiding effect in early search stages. The approach also incorporates adaptive dynamic modification of pheromone evaporation and the 3-opt metaheuristic to avoid local optima. Simulation results on TSP datasets and engineering applications show significant improvements. In [9], a Dynamic Adaptive ACO algorithm (DAACO) is proposed. DAACO dynamically adjusts the number of ants to maintain search diversity and applies a hybrid local selection strategy. Results demonstrate superior performance over conventional ACO in convergence time, optimization quality, and average solution value. In [38], ACO is applied to optimize fertilizer distribution logistics at PT. Socfindo Bangun Bandar, modeling six distribution points as a weighted graph. With only one iteration, (NC = 1), the optimized route reduced the travel distance from 16.10 km to 15.71 km. This highlights ACO's effectiveness in improving agricultural logistics efficiency.

5. Proposed Method

The Ant Colony Optimization (ACO) algorithm is one of the many approaches used to solve the Travelling Salesman Problem (TSP) due to its ability to achieve near-optimal solutions using swarm intelligence. ACO, however, faces several issues like convergence, stagnation, and inefficiency for large TSP problems. To improve upon these shortcomings, we design an adaptive active learning ACO variant that incorporates metaheuristic refinement strategies using adaptive and divide-and-conquer techniques to boost algorithm efficiency. Although sub-tours (clusters of cities) are dealt with independently in the Divide-and-Conquer technique to handle large-scale TSP difficulties using ACO, they must still be linked together to form a complete tour. This is when bridging heuristics go in role.

Our approach has four improvements as follows:

1. Enhanced Pheromone Update Rules – Our novel contribution is hyper adjusted pheromone limits that define both the upper and lower pheromone bounds to enhance exploration and avoid premature convergence.
2. Adaptive Strategies – We apply ant colony partitioning where ants are separated in subgroups where every subgroup implements different search patterns to promote diversity and avoid stagnation.

3. Enhanced Initialization – We adopt Smart Initialization that uses a heuristic like the nearest neighbor's method for an initial assignment of the pheromones which makes the system's convergence faster.
4. Dynamic and Big Scale Modifications of the TSP – One of the ways to deal with larger TSP problems is to break them down into more manageable pieces, resolve them separately, and then combine the results efficiently considering the divide-and-conquer technique.

At every time it runs, the algorithm determines whether a predefined ending condition was reached. A convergence check is based on two parameters: (1) completing the maximum number of iterations, and (2) seeing no difference in the best-found tour length for a set number of sequential iterations. If either condition is met, the algorithm finishes as provides the best solution found. Alternatively, the procedure resumes to the solution construction phase and continues to iterate. This method prevents needless computations after the algorithm has stabilized, ensuring effective use of computational resources. Blended together these modifications are intended to create greater balance between exploration and exploitation, refinement of the provided solution and scalability for approaching more complicated large TSP models is achieved.

Improved ACO algorithm

Step 1. Initialization

Input number of antes (m), cities (n), pheromone matrix τ, heuristic β, evaporation rate ρ.

Initiate pheromone distribution (τ_0) using the nearest neighbor as heuristic by

$\alpha(i,j) = 1/(n * \frac{1}{n*L(n)}$,where L(n) is the nearest neighbor.

Input pheromone limits (τ_min, τ_max).

Step 2. Adaptive Strategies

Split (m) using different search techniques to:

- Use randomization to find various new path.
- Chose the new path with highest pheromone amount.
- Provide a balance between exploration and exploitation for ants.

Step 3. Solution process in Parallel

For each (m):

3.1. Select first city randomly from (n).

3.2. Create a path by chose the next city (j) form (n) depending on probability from eq.(1) using pheromone intensity and heuristic information.

Step 4. Update Pheromone

4.1. update pheromone value using Eq. (2).

4.2. Execute the Max-Min Ant System for strengthening the best path while deploying elitist ants to leave additional pheromones on the entire best path.

4.3. Limit all pheromone values to [τ_min, τ_max]. To avoid stagnation.

Step 5. Divid and conquer for large-scale TSP:

5.1. if TSP is large instance.
5.1.1. Divide cities into smaller area.
5.1.2. Execute ACO to each cluster independently.
5.1.3. Combine solutions by applying bridging heuristic, to create a comprehensive solution.

Step 6. Convergence Check

6.1 If stopping conditions have been achieved. (e.g., Maximum iterations or no change.), return the best path.
6.2 Otherwise, go to Step 2.

Step 7. Return: Best tour length, average tour length, standard deviation, and total execution time.

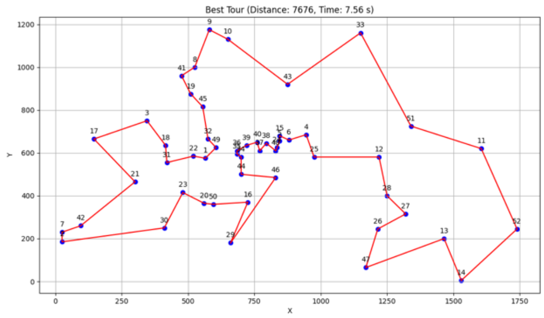
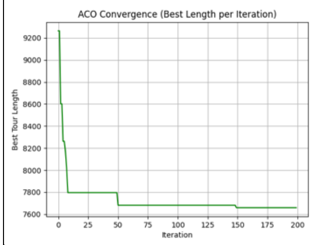
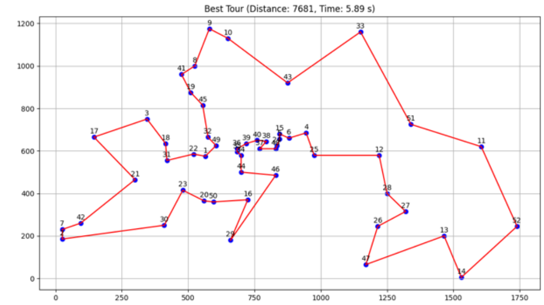
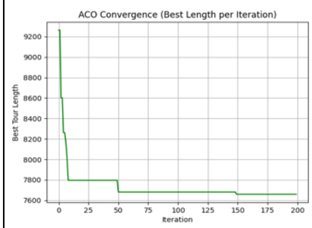
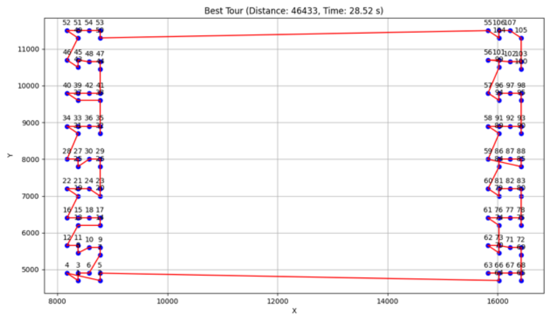
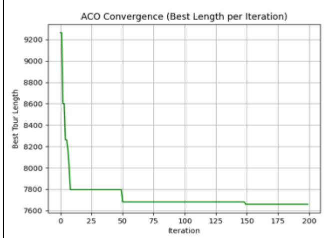
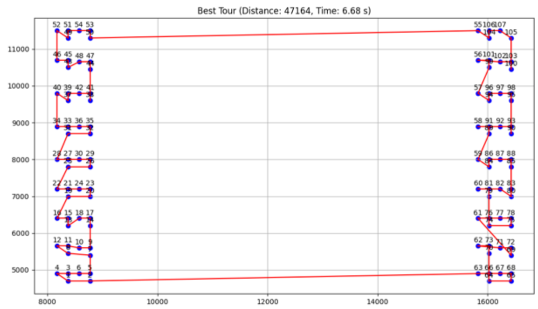
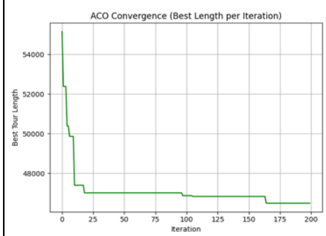
6. Evaluation

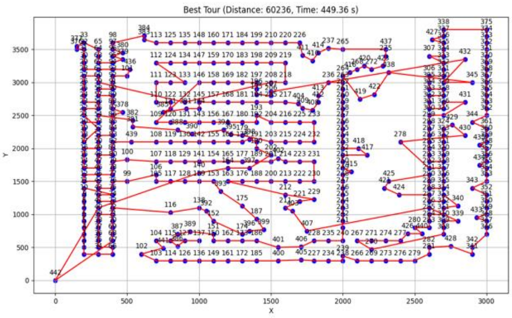
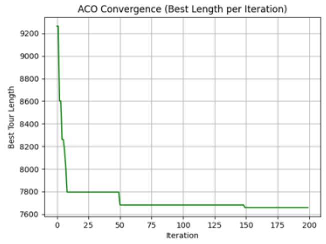
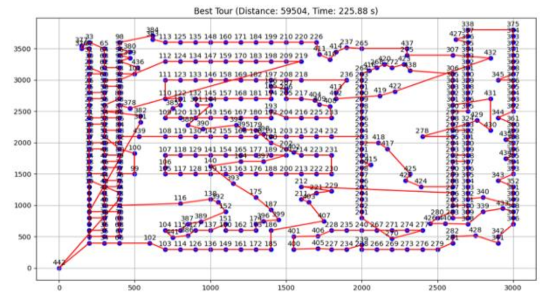
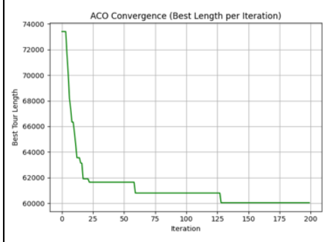
A number of experiments on standard TSP instances from the TSPLIB dataset were used to assess the performance of the proposed Improved ACO algorithm. The experiments were written in Python and performed out on a system with an Apple M1 processor, 16 GB of RAM, and macOS as the operating system. We chose the following TSP instances to reflect small-, medium-, and large-scale issues. For small, chosen is berline52, which has 52 cities; for medium, chosen is pr107, which has 107 cities; and finally, for a large number of cities, chosen is pcb442, which has 442 cities with iteration equal to 150 because it takes many times in execution. The parameters for both the traditional ACO and the improved version are number of ants: 50; max iteration: 200; α equal to 1.0; β equal to 5.0; and ρ equal to 0.5. In order to maintain consistency, each experiment was conducted 30 times, recording the best tour length, average tour length, standard deviation, and time taken for each method.

Table 1. proposed work result
(best tour, average length, std. dev. and time)

Instance	Algorithm	Best Tour	Avg. Tour	Std. Dev	Time (s)
berlin52	Standard ACO	7677	9367	617	26.96
	Improved ACO	7681	9336	542	5.89
pr107	Standard ACO	46433	56701	5292	28.52
	Improved ACO	47164	56738	5110	6.68
pcb442	Standard ACO	60236	71350	3744	449.36
	Improved ACO	59504	71025	3839	225.88

Table 2. The result of the proposed work with corresponding convergence

Dataset	Standard ACO	Convergence
berlin52 Standard ACO		
berlin52 Improved ACO		
Pr107 Standard ACO		
Pr107 Improved ACO		

Pcb442 Standard ACO		
Pcb442 Improved ACO		

6.1 Convergence Analysis.

To analyze the convergence behavior, we showed the mean tour duration within iterations between the two algorithms. The Improved ACO consistently achieves faster convergence with to its creative initialization and adaptable methods, permitting it to produce high-quality solutions with less iterations.

6.2 Discussion

On the pr107 instance, the Standard ACO had a better best tour (46433 vs. 47164). Both algorithms yielded similar average tour lengths (~56700), with the Improved ACO having a lower standard deviation (5110 vs. 5292). The execution time was lowered by more than four times, from 28.52s to 6.68s. This demonstrates that, while the best answer may decline slightly in some circumstances, the enhanced version preserves consistency while greatly reducing calculation time. In the pcb442 example, the Improved ACO surpassed the Standard ACO in both the best (59504 vs. 60236) and average tours. Although the standard deviation increased slightly (3744 → 3839), this is acceptable given the greater overall quality and the significant drop in time from 449.36s to 225.88s (almost twice as fast). In the pcb442 example, the Improved ACO surpassed the Standard ACO in both the best (59504 vs. 60236) and average tours. Although the standard deviation increased slightly (3744 → 3839), this is acceptable given the greater overall quality and the significant drop in time from 449.36s to 225.88s (almost twice as fast).

The Improved ACO consistently reduces execution time, resulting in 2 times to 4.5 time quicker performance depending on the instance. In two of the three situations, it also decreased the standard deviation, indicating more consistent convergence behavior.

While the pr107 case sacrifices some best tour quality, the average performance and reduced variability provide compensation for it.

7.Conclusion

This work proposed an improved Ant Colony Optimization (ACO) algorithm for solving the Traveling Salesman Problem (TSP) more efficiently and consistently. The revised ACO shows considerable performance benefits thanks to a range of modifications, including parallel solution generation, optimized pheromone updating techniques, and increased computing efficiency. Experimental results on standard TSPLIB datasets (berlin52, pr107, and pcb442) show that the proposed method achieves faster convergence, reduced execution time (up to 4.5x faster), and more consistent solutions with lower standard deviation in most situations. Although the best tour duration was somewhat reduced in one case, the trade-off is exceeded by overall improvements in stability and speed, particularly for larger-scale problems. These findings indicate that the proposed enhancements make ACO a more scalable and realistic solution to real-world, large-instance combinatorial optimization issues. Future work will investigate adaptive parameter tweaking, integration with local search algorithms, and more divide-and-conquer tactics to improve performance and scalability.

References

[1] S. C. Y. S. a. L. Z. Xinyi Shu, ""Ant colony algorithm in the application of path planning research"," China, 2024.

[2] T. M. L. a. P. D. G. António Agrela Freitas, "Ergonomic Risk Minimization in the Portuguese Wine Industry: A Task Scheduling Optimization Method Based on the Ant Colony Optimization Algorithm," *Processes,* vol. 10, no. 1364, pp. 1-22, 2022.

[3] A. Lu, "An ant colony optimization algorithm based on the chaos search," in *Journal of Physics: Conference Series* , USA, 2020.

[4] S. B. N. B. M. S. Stefan Bock, "A survey on the Traveling Salesman Problem and its variants in a warehousing context," *European Journal of Operational Research,* vol. 322, no. 1, pp. 1-14, 2024.

[5] Y. B. C. P. G. Yimeng Min, "Unsupervised Learning for Solving the Travelling Salesman Problem," in *37th Conference on Neural Information Processing Systems*, USA, 2023.

[6] O. V. Z. R. P. K. T. A. V. a. I. V. H. A V Riabko, "Algorithm of ant colony optimization (ACO) for 3D variation traveling salesman problem," in *Journal of Physics, XIV International Conference on Mathematics, Science and Technology Education*, Ukraine, 2022.

[7] E. A. I. I. M. a. L. S. Blagoveshchenskaya, "Blagoveshchenskaya, Ekaterina A., Ilya I. Mikulik and Lutz Strüngmann. "Ant colony optimization with parameter update using a genetic algorithm for travelling salesman problem.”

(2021).," in *Proceedings of the Workshop "Models and Methods for Researching Information Systems in Transport 2020"*, Russia, 2020.

[8] W. Gao, "New Ant Colony Optimization Algorithm for the Traveling Salesman Problem," *International Journal of Computational Intelligence Systems,* vol. 13, no. 1, pp. 44-55, 2020.

[9] A. L. W. L. P. G. Huijun Liu, "DAACO: adaptive dynamic quantity of ant ACO algorithm to solve the traveling salesman problem," *Complex and Intelligent Systems,* vol. 9, p. 4317–4330, 2023.

[10] A. L. T. ,. M. M. F. a. R. V. O. Ivohin E. V., "About Of The Annealing Method Using For The Traveling Salesman Problem Solution With The Fuzzy Time Perception," *Radio Electronics, Computer Science, Control.,* vol. 4, no. 1, pp. 56-63, 2024.

[11] D. Terzi, "A Natural Approach To Solving The Traveling Salesman Problem," *Cybernetics And Computer Technologies,* vol. 4, no. 1, pp. 43-52, 2023.

[12] G. E. A. M. W. A.-N. A. H. A. Warif B. Yahia, "Traveling Salesman Problem Methods of Solution Survey," *International Journal of Psychosocial Rehabilitation,* vol. 24, no. 5, pp. 8565-8581, 2020.

[13] M. Yu, "A Solution Of Tsp Based On The Ant Colony Algorithm Improved By Particle Swarm Optimization," *Discrete And Continuous Dynamical Systems Series S,* vol. 12, no. 4, p. 979–987, 2019.

[14] O. R. M. G. Stefka Fidanova, "Ant Colony Optimization Algorithm for Fuzzy Transport Modelling," *Computer Science and Information Systems,* vol. 21, no. 1, p. 237–240, 2020.

[15] E. C. a. S. K.-U. Erfan Babaee Tirkolaeea, "Traveling salesman problem with drone and bicycle: multimodal last-mile e-mobility," *International Transactions In Operational Research,* vol. 0, no. 13452, pp. 1-27, 2024.

[16] A. M.-V. L. T. J. T. María De los Ángeles Cosío León, "A Variant of Parallel-Hybrid Genetic Algorithm for Large-Scale Traveling Salesman Problem," in *Handbook of Formal Optimization.*, Singapore, Springer, 2023, pp. 1-20.

[17] R. M. H. P. L. Vitoria Pureza, "Modeling And Solving The Traveling Salesman Problem With Priority Prizes," *SciELO,Brazilian Operations Research Society,* vol. 38, no. 3, pp. 499-522, 2018.

[18] ,. M. S. A. Md. Limonur Rahman Lingkona, "Application of an improved ant colony optimization algorithm of hybrid strategies using scheduling for patient management in hospitals," *Heliyon,* vol. 10, no. 1, pp. 1-18, 2024.

[19] L. T. Rasheed, "Optimal Tuning of Linear Quadratic Regulator Controller Using Ant Colony Optimization Algorithm for Position Control of a Permanent Magnet DC Motor," *IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING,* vol. 20, no. 3, pp. 29-41, 2020.

[20] M. D. a. T. Stützle, "Ant Colony Optimization: Overview and Recent Advances," in *Handbook of Metaheuristics*, Belgium, Springer, Cham, 2019, pp. 311-351.

[21] L. C. K. M. M. M. A. S. Kho, "PropositionalSatisfiability Logicvia Ant ColonyOptimization in Hopfield NeuralNetwork," *Malaysian Journal of Mathematical Sciences,* vol. 16, no. 1, pp. 37-53, 2022.

[22] JuanCheng, "DynamicPathOptimizationBasedonImprovedAnt ColonyAlgorithm," *Journal of Advanced Transportation, Hindawi,* vol. 2023, no. 1, pp. 1-11, 2023.

[23] A. K. Mandal, "A Step-By-Step Mathematical Derivation Of Ant Colony Optimization Algorithm For Solving The Traveling Salesman Problem," *International Research Journal Of Modernization In Engineering Technology And Science,* vol. 5, no. 12, pp. 1902-1917, 2023.

[24] Y. Wan, "Simulated Annealing Algorithm Use for Assessing Administrative Level in Colleges and Institutions," *International Journal of Decision Support System Technology,* vol. 16, no. 1, pp. 1-18, 2024.

[25] A. Toshev, "Particle Swarm Optimization and Tabu Search Hybrid Algorithm for Flexible Job Shop Scheduling Problem – Analysis of Test Results," *CYBERNETICS AND INFORMATION TECHNOLOGIES ,* vol. 19, no. 4, pp. 26-44, 2019.

[26] T. S. S. B. a. M. H. Olga Speck, "The Basics of Evolution Strategies: The Implementation of the Biomimetic Optimization Method in Educational Modules," *biomimetics,* vol. 9, no. 439, pp. 1-27, 2024.

[27] A. S. Akopov, "An Improved Parallel Biobjective Hybrid Real-Coded Genetic Algorithm with Clustering-Based Selection," *CYBERNETICS AND INFORMATION TECHNOLOGIES,* vol. 24, no. 2, pp. 32-49, 2024.

[28] A. K. U. N. a. S. M. Saira Beg, "Performance Evaluation of Bionomic Algorithm (BA) in Comparison with Genetic Algorithm (GA) for Shortest Path Finding Problem," *IJCSI International Journal of Computer Science Issues,* vol. 8, no. 6, pp. 238-242, 2011.

[29] L. M. G. a. F. d. L. V. Maniezzo, "Ant colony optimization," in *New Optimization Techniques in Engineering,*, Berlin, Springer, 2004, pp. 101 - 121 .

[30] S. M. V. a. A. S. A. Rezvanian, "An overview of ant colony optimization algorithms for dynamic optimization problems," in *Optimization Algorithms: Classics and Recent Advances*, United Kingdom, IntechOpen, 2023, pp. 1-19.

[31] R. S. L. H. C. Salsabillah Hazizah, "Ant Colony Optimization Algorithm For Traveling Salesman Problem In Distributing Fertilizer," *Jurnal Matematika Dan Pendidikan Matematika,* vol. 8, no. 2, pp. 345-358, 2023.

[32] O. B.-H. A. A. S. A. A. Oliazadeh, "Ant Colony Optimization Algorithms: Introductory Steps To Understanding," In *Computational Intelligence For Water And Environmental Sciences*, Singapore, Springer, 2022, pp. 137-154.

[33] M. R. A.-T. H. Ashour W., "Optimization of Traveling Salesman Problem based on Adaptive Affinity Propagation and Ant Colony Algorithms," *International Journal of Computer Applications,* vol. 181, no. 29, p. 25–31, 2018.

[34] N. F. P. T. S. Bib Paruhum Silalahi, "Use of Ant Colony Optimization Algorithm for Determining Traveling Salesman Problem Routes," *Jurnal Matematika MANTIK,* vol. 5, no. 2, pp. 100-111, 2019.

[35] Y. L. ,. A. L. Q. H. N. Z. N. Q. M. Z. A. L. C. MEIJIAO LIU, "A Slime Mold-Ant Colony Fusion Algorithm for Solving Traveling Salesman Problem," *IEEE,* vol. 8, no. 1, pp. 202508-202521, 2020.

[36] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem," *Engineering Applications of Artificial Intelligence,* vol. 92, pp. 1-10, 2020.

[37] R. Skinderowicz, "Improving Ant Colony Optimization Efficiency for Solving Large TSP Instances," *Skinderowicz, Rafał. "Improving Ant Colony Optimization efficiency for solving large TSP instances." Applied Soft Computing,* vol. 120, no. 1, pp. 108653-108681, 2022.

[38] X. W. L. Z. Y. Z. a. L. C. xTeng Fei, "Research on improved ant colony optimization for traveling salesman problem," *Mathematical Biosciences and Engineering,* vol. 19, no. 8, pp. 8152-8186, 2022.

[39] Network, H., 2019. Artificial Bee Colony for Minimizing the Energy Consumption in Mobile Ad. *Advances on Computational Intelligence in Energy: The Applications of Nature-Inspired Metaheuristic Algorithms in Energy*, p.21.

[40] Tareq, M., Sundararajan, E.A., Harwood, A. and Bakar, A.A., 2021. A systematic review of density grid-based clustering for data streams. Ieee Access, 10, pp.579-596.

[41] Abdelhaq, M., Alsaqour, R., Algarni, A., Alabdulhafith, M., Alawi, M., Taha, A., Sharef, B. and Tariq, M., 2020. Human immune-based model for intrusion detection in mobile ad hoc networks. Peer-to-Peer Networking and Applications, 13, pp.1046-1068.

[42] Tareq, M.U.S.T.A.F.A. and Sundararajan, E.A., 2021. An evolving approach to data streams clustering based on Chebychev with false merging. Journal of Theoretical and Applied Information Technology, 99(9).

[43] Tareq, M. and Sundararajan, E.A., 2020. A new density-based method for clustering data stream using genetic algorithm. Technology Reports of Kansai University, 62(11), p.16.

[44] Tareq, M., Mohammed, M.J. and Rashid, O.F., 2022. Text Cryptography Based on Three Different Keys. AL-MANSOUR JOURNAL, (37).

[45] Tareq, M.U.S.T.A.F.A. and Sundararajan, E.A., 2021. An evolving approach to data streams clustering based on Chebychev with false merging. Journal of Theoretical and Applied Information Technology, 99(9).

[46] Farhan, Y.H., Shakir, M., Abd Tareq, M. and Shannaq, B., 2024. Incorporating Deep Median Networks for Arabic Document Retrieval Using Word

Embeddings-Based Query Expansion. Journal of Information Science Theory and Practice, 12(3), pp.36-48.

[47] Abd, M.T., Mohd, M. and Abd, M.T., 2018, March. Investigation of Data Representation Methods with Machine Learning Algorithms for Biomedical Named Enttity Recognition. In 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP) (pp. 1-6). IEEE.

[48] Tareq, M.U.S.T.A.F.A. and Sundararajan, E.A., 2021. An evolving approach to data streams clustering based on Chebychev with false merging. Journal of Theoretical and Applied Information Technology, 99(9).

[49] Papadopoulos, M., Mavromoustakis, C.X., Skourletopoulos, G., Mastorakis, G. and Pallis, E., 2014, July. Performance analysis of reactive routing protocols in mobile ad hoc networks. In 2014 International Conference on Telecommunications and Multimedia (TEMU) (pp. 104-110). IEEE.

[50] Rashid, O.F., Mohammed, M.J. and Tareq, M., 2022. Text Cryptography Based on Three Different Keys. Al-Mansour Journal, 37(1), pp.17-26.

[51] Farhan, Y.H., Shakir, M., Abd Tareq, M. and Shannaq, B., 2024. Incorporating Deep Median Networks for Arabic Document Retrieval Using Word Embeddings-Based Query Expansion. Journal of Information Science Theory and Practice, 12(3), pp.36-48.