International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(4):542-553 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i04.01134

An Effective Route Tracing Using Enhanced Ant Colony Optimization Techniques: Travelling Salesman Problem

S Selvi*, D Dhinoovika, R Harshana, R Muthulakshmi

Department of Computer Science and Engineering, Government College of Engineering, Bargur, Tamil Nadu 635104, India. *Corresponding Author's Email: sel_raj241@yahoo.com

Abstract

The Travelling Salesman Problem (TSP) is a well-known optimization that determines the shortest route that visits a particular set of towns and returns to the start line. As an NP-hard, its complexity will increase considerably as the number of cities increases. Several heuristic algorithms, such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithm (GA), have been developed to tackle this problem efficiently. This work establishes an approach that employs a multi-goal optimization algorithm that optimizes for minimizing the tour distances and maximizing the diversity of the towns visited. This paper aims to apply an enhanced ACO with conventional GA and PSO algorithms separately to leverage the strengths of both algorithms. The PSO and GA paths enhance the pheromone in global exploration, which speeds up the pheromone initialization of ACO. Thus, optimized routes by ant colony get integrated with pheromone reinforcement to obtain enhanced pheromone in the hybrid of GA-ACO and PSO-ACO. The ACO algorithm applies enhanced state transition to progress the search using the angle guidance function, and the best pheromone level improves convergence speed of the enhanced algorithms. This project has been designed as an intuitive and user-friendly interface for users to input TSP instances, select optimization algorithms, and visualize the solutions generated by each algorithm. Experiments are done to assess overall performance of ACO, GA-ACO, and PSO-ACO based on the best solution and convergence speed. Lastly, the effectiveness and performance of those algorithms in solving TSP across diverse scenarios and input parameters are compared.

Keywords: Ant Colony Optimization, Multi-Objective Algorithm, Optimization Algorithm, Route Tracing, Travelling Salesman Problem.

Introduction

The main objective of TSP is to optimize the distance covering the complete route travelled at some point in the journey. The Ant Colony System (ACS) has been proposed to solve the TSP (1); also, for the asymmetric traffic assignment problem, the algorithm's traffic assignments are compared to known solutions. ACO algorithm performed well due to its heuristic exploitation. Still, the asymmetric nature of the problem adds complexity, affecting efficiency and scalability (2). Another application of ACO is for integrated production and distribution scheduling, which aims for holistic and efficient schedules (3). The PSO approach for the shortest path problem highlights PSO's exploration capabilities but lacks premature convergence issues (4).

A logistics terminal distribution mode optimization using the ant colony algorithm produced the best simulation results, improving logistics efficiency but facing parameter tuning challenges (5). The ACO algorithm has been implemented for job shop scheduling with time windows, showing adaptability that is both efficient and effective (6). For finding the shortest path routing, the genetic algorithm has shown promising results but needs significant computational resources and careful tuning (7). Project scheduling and resource allocation have been explored with ACO, which has shown effectiveness in managing complex project scenarios (8).

PSO approach for shortest path planning, which highlights its efficiency and challenges in sensitivity to parameters (9). Genetic algorithms for route finding require expertise in tuning and face computational costs (10). Researchers have applied an ant-inspired algorithm for vehicle routing, which has proven effective in exploring solution space (11). An algorithm for IT task scheduling, addressing practical IT sequencing problems with multiple objectives, has been proposed and shown the best results compared

(Received 19th May 2024; Accepted 21st October 2024; Published 30th October 2024)

with simulated annealing, genetic algorithm, tabu search, and neural network algorithms (12).

An in-depth examination of particle swarm optimizers has shown remarkable in the theoretical foundations, algorithmic aspects, applications, and highlighting strengths and limitations (13). Add ACO, a modified ACO algorithm for TSP, has been developed, aiming to enhance solution quality but facing challenges in complexity and parameter sensitivity (14). A hybrid algorithm for TSP, leveraging genetic algorithms and reinforcement learning, has been proposed, with proven potential for improving solutions in terms of quality but facing challenges in implementation and parameter tuning (15). Various purposes of particle swarm optimization methods, emphasizing simplicity and effectiveness, have been presented (16), addressing issues and challenges for future research perspectives (17).

A novel hybrid approach known as the Genetic Simulated Annealing Ant Colony machine with Particle Swarm Optimization techniques (GSA-ACO-PSO) integrates the strengths of ACO for preliminary solution technology, Simulated Annealing for enhancing those solutions, and PSO for sharing pheromone statistics between groups (18). These algorithms, such as ACO, PSO, and GA, have shown stable exploration and exploitation, adapt dynamically, and are, without difficulty, parallelized for big-scale issues, often taking advantage of hybridization for improved performance (19). ACO's effectiveness in solving real-international troubles, including journeying salesman trouble and car routing problems, underscores its broad applicability. By delving into modern-day challenges and outlining destiny research guidelines, this evaluation aims to deepen know-how and foster improvements in ACO, supplying treasured insights to researchers and practitioners alike (20). TSP is applicable in various fields like manufacturing, transportation, logistics, energy dispatching (21), and telecommunications, wherein minimizing journey distances or prices is essential. This survey explores various optimization strategies focused on enhancing existing approaches to further research.

Methodology

Motivation

TSP is NP-hard, in which the problem's complexity grows exponentially because the range of cities increases. Given its complexity, various optimization strategies and algorithms are employed to find approximate solutions, including metaheuristic processes such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Genetic Algorithms (GA). These strategies are specifically effective in handling large-scale TSPs and are widely applied throughout various fields due to their capability to provide near-top-rated solutions efficiently.

Hence, this paper aims to integrate fundamental PSO and GA algorithms with enhanced ACO, thereby enhancing GA-ACO and PSO-ACO leverages the strengths of both algorithms. The GA and PSO trails acquire higher pheromones in global exploration, runs the pheromone initialization of ACO. Thus, optimized routes by ants get unified with pheromone support to obtain enhanced pheromones in the hybrid PSO-ACO and GA-ACO. Here, two enhancements in the ACO algorithm are applied: a) Enhanced state transition using angle guidance function, and b) best pheromone level (updated rule). The performance of the enhanced GA-ACO and PSO-ACO algorithms have been compared. Experimental results conclude that a concise and precise route tracing scheme is introduced, primarily based on the enhanced ACO algorithm that progresses the search and improves the enhanced PSO-ACO convergence speed.

Pseudocode

The pseudocode of the ACO and variants of ACO i.e. GA-ACO and PSO-ACO are explained here under.

Ant Colony Optimization

Step 1: Initialize pheromone levels τ_{ij} on each path, heuristic function η_{ij} and other parameters such as ρ, α, β, t_{max} , L best, and the number of steps n and ants count m;

Step 2: For each ant k from 1 to m: Repeat until all cities are visited;

Step 3: Calculate the probability of moving from city by Eq. [1];

Step 4: Use roulette wheel selection to choose the next city based on probabilities p_{ij}^{k}.Move ant k to the selected city;

Step 5: Evaporate pheromone on all edges as Eq. [2];
Step 6: Update pheromone by Eq. [3] based on ant's tours;
Step 7: If a shorter tour is found: Update L as best and record the tour;
Step 8: Repeat steps 2-4 for t_{max} iterations or till a termination condition is obtained.

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l \epsilon allowed} \ [\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}} \quad [1]$$

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) \quad [2]$$

$$\tau_{ij}(t+1) = \tau_{ij} \sum_{k=1}^{m} \ \Delta\tau_{ij}^{k} \quad [3]$$

where $\Delta\tau_{ij}^{k} = \{\frac{Q}{L_k} \quad if\ ant\ k\ uses\ edge\ i \rightarrow j\ in\ its\ tour\ 0 \quad otherwise$

Enhanced ACO with Genetic Algorithm

Step 1: Initialize the population size N, chromosome length, and other parameters.
Step 2: Compute the fitness value of an individual in the population by Eq. [4];
Step 3: Each individual retains its ideal fitness value $G_{best,}$ and the present global ideal fitness value H_{best};
Step 4: Update the position of an individual using Eq. [5];
Step 5: Repeat steps 2-4 until an exit condition is met;
Step 6: Transform the peak route by GA into the pheromone concentration for the ACO algorithm;
Step 7: Initialize the pheromone levels on each path and set other parameters for ACO;
Step 8: Randomly distribute ants to each other city;
Step 9: An individual ant chooses the subsequent location based on probability Eq. [6] for enhanced state transition (21);
This enhanced state transition improves the route search diversity, which is obtained by integrating the angle guidance function $[\mu_{ij}(t)]^{\gamma}$ thereby increasing the route search efficiency. Simultaneously, the heuristic function $\eta'_{ij}(t)$ gets optimized by introducing distance from the succeeding likely node to the nearby node built on the conventional heuristic function $\frac{1}{(\lambda.d_{ij}+\mu.d_{jg})2}$. Hence, the accuracy of the route search is improved.
Step 10: Compute the distance traveled by individual ant and update the ideal path of this iteration.
Step 11: Modify the best pheromone levels on each path according to Eq. [7];
Step 12: Check if the loop condition is met. Otherwise, go to step 9. If affirmative, output the ideal path, concluding the algorithm.

The integration of the GA-ACO algorithm strengthens the pheromone accumulation of the ants' colony at the initial stage. Meanwhile, the ACO gets improvised based on the GA route, evading the ACO's blind search.

Enhanced ACO with PSO Algorithm

Step 1: Set the particle position, speed, and population size N.
Step 2: Compute the fitness value of each particle as per Eq. [8];
Step 3: The individual particle retains its optimal fitness value, $A_{best,}$ and the current global optimal fitness value, B_{best}, correspondingly. The particle consistently adapts its status to explore alternative solutions through Eq. [9];
Step 4: Check if the condition is met; if affirmative, output the ideal path achieved by the particle swarm else go to step 2;
Step 5: Convert the most efficient direction found by the Particle Swarm Optimization into pheromone ranges for the Ant Colony Optimization algorithm, establishing the pheromone attention to 1.3;
Step 6: The ant colony is distributed at random to each city;
Step 7: Every ant chooses the succeeding location depending on Eq. [10] for enhanced state transition (21);
This enhanced state transition improves the route search diversity, which is obtained by integrating the angle guidance function $[\mu_{ij}(t)]^{\gamma}$ thereby increasing the route search efficiency. Simultaneously, the heuristic function $\eta'_{ij}(t)$ gets optimized by introducing distance from the succeeding likely node to the nearby node built on the conventional heuristic function $\frac{1}{(\lambda.d_{ij}+\mu.d_{jg})2}$. Hence, the accuracy of the route search is improved.
Step 8: Compute the distance traveled by every ant,

verify the globally optimal path of the loop, and change the table.

$$d_{ij} = \sqrt{(m_i - m_j)^2 + (n_i - n_j)^2} \quad [4]$$

$$v_{i+1} = wv_i + c_1 r_1 (G_{best} - m_i) + c_2 r_2 (H_{best} - m_i) \quad [5]$$

$$A'_{ij}(t) = [\tau'_{ij}(t)]^{\alpha} [\eta'_{ij}(t)]^{\beta} [\mu_{ij}(t)]^{\gamma} \quad [6]$$

$$\tau'_{ij}(t+n) = (1-\rho').\tau'_{ij}(t) + \Delta\tau'_{ij}(t) \quad [7]$$

where $\Delta\tau'_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij(t)}^{k}$

$$Fitness = \sum_{(i,j)\subset particle} d_{ij} \quad [8]$$

where $d_{ij} = \sqrt{(p_i - p_j)^2 + (q_i - q_j)^2}$

$$v_{i+1} = wv_i + c_1 r_1 (A_{best} - p_i) + c_2 r_2 (B_{best} - p_i) \quad [9]$$

$$p_{i+1} = p_i + v_{i+1}$$

$$A'_{ij}(\mathrm{t}) = \{ \frac{[\tau' ij(t)]^{\alpha} [\eta' ij(t)]^{\beta} [\mu_{ij}(t)]^{\gamma}}{\sum_{s\subset Tallowed\ ,a} [\tau' ij(t)]^{\alpha} [\eta' ij(t)]^{\beta} [\mu_{ij}(t)]^{\gamma}} \ 0 \quad [10]$$

$$\tau'_{ij}(t+n) = (1-\rho').\tau'_{ij}(\mathrm{t}) + \Delta\tau'_{ij}(t) \quad [11]$$

where $\Delta\tau'_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$

here, ρ' depends on the adjustment coefficient whose value is less than 1, the optimal route multi-objective value, and the distance from the source region to the target region. The optimal route multi-objective value reaches a minimal value comparatively than at the initial stage of the algorithm, which is ahead of the identified path. This ants' capability maximizes the speed of the ant colony search, thus making immediate convergence. Subsequently, the resultant best pheromone level increases the ant colony concentration on the identified optimal path to attain global optimization, increasing convergence speed.

Step 10: Check if the condition is satisfied; else, go to step 7. If affirmative, yield the optimal path, concluding the algorithm.

The integration of the PSO-ACO algorithm strengthens the pheromone accumulation of the ants' colony at the initial stage. Meanwhile, the ACO gets improvised based on the PSO route, evading the ACO's blind search. This leads to improving the PSO-ACO algorithm search and convergence speed.

Step 9: Adjust the best pheromone levels (21) on individual path according to Eq. [11];

Figure 1 gives the corresponding flowcharts of the enhanced GA-ACO and PSO-ACO algorithms and depicts their pseudocode.

Computational Complexity

Computational complexity is a set of measures that includes time and space complexity, which determines how many resources are utilized by the operations specified in the algorithms. These complexity measures can be approximated by big-*O* notation. Table 1 describes the time complexity *T(n)* of the ACO and variants of ACO algorithms.

Table 1 clearly demonstrates that the time complexity of each algorithm is based on various factors like the number of iterations, particles, ants, and cities, and mainly on the problem's structure and requirements. Each algorithm's complexity is induced by the problem's size and specific operations like fitness evaluation or pathfinding. This paper compares the experimental results of optimization algorithms evaluated based on above factors.

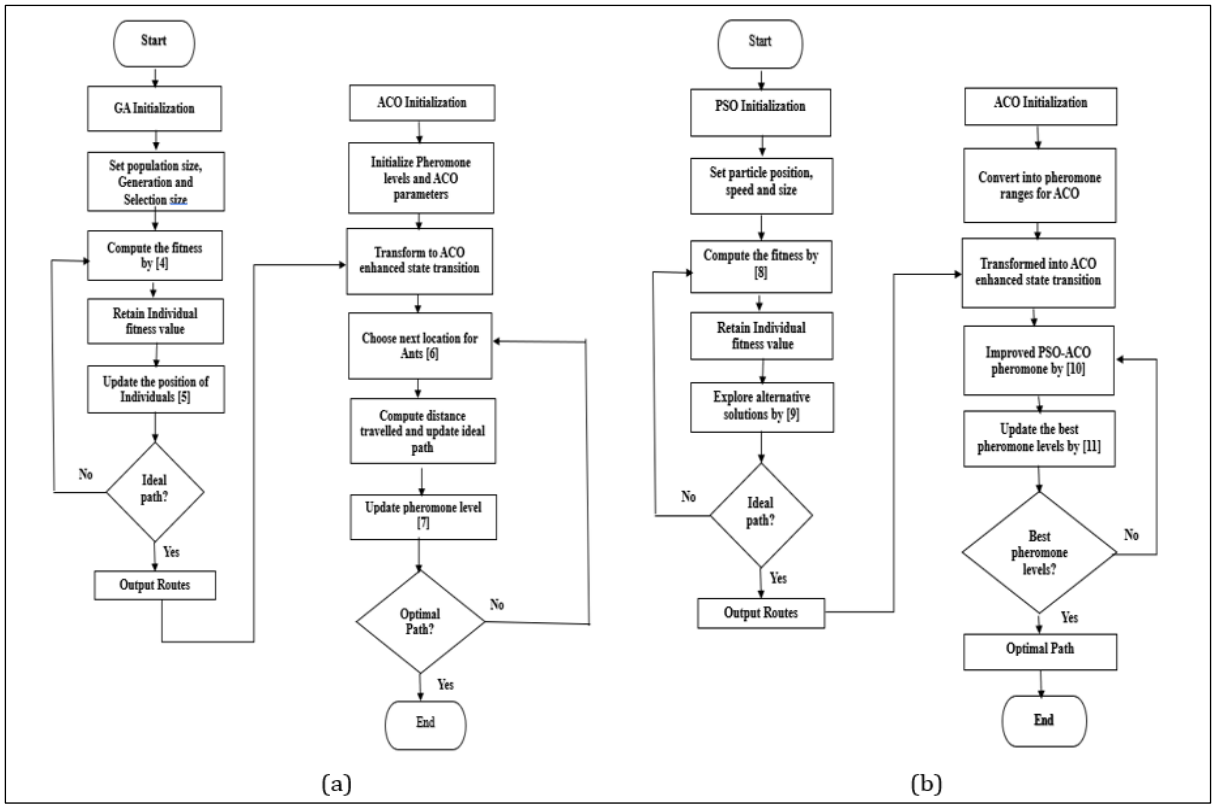


Figure 1: Flowchart of enhanced (a) GA-ACO algorithm (b) PSO-ACO algorithm

Table 1: Description of Time Complexity of the ACO and Variants of the ACO Algorithm

Name of the Algorithms	Time Complexity T(n)	Parameters Description
Particle Swarm Optimization (PSO) Algorithm	$O\ (T*N*(D+C))$	T: Quantity of iterations. N: Number of particles. D: Dimensionality of the problem. C: Complexity of the health evaluation feature.
Ant Colony Optimization (ACO) Algorithm	O (T∗M∗N^2)	T: Range of iterations. M: Number of ants. N: Range of cities.
Genetic Algorithm (GA)	O (G∗P∗L ^2)	G: A variety of generations. P: Population Size. L: Length of each chromosome or course. F: Fitness assessment complexity, which relies upon the trouble.
Genetic Algorithm - Ant Colony Optimization (GA-ACO) Algorithm	O (P∗D∗G∗M∗D∗I)	P: Population Size. G: A variety of generations. D: Number of decision variables. I: Range of iterations. M: Number of ants.
Particle Swarm Optimization - Ant Colony Optimization (PSO-ACO) Algorithm	$O\ (T_{PSO} * N_{PSO} * D) + O\ (T_{ACO} * M_{ACO} * E)$	T: Quantity of iterations. N: Number of particles. D: Dimensionality of the problem. M: Number of ants. E: Number of Edges in the Graph.

Results and Discussion

The hybrid approach is written in Python script run on the machine set up with Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz 2.11 GHz, Windows 11 Pro machine, and 16 GB of RAM. This hybrid strategy is implemented across distinct datasets, specifying distinct topological systems. Implementing the ACO, GA-ACO, and PSO-ACO algorithms starts with initializing different parameters in the script. Table 2 shows the various parameters initialization for ACO and variants of the ACO algorithm

Table 2: List of Parameters Initialization of ACO and Variants of ACO Algorithm

Parameter	ACO	GA-ACO	PSO-ACO
Population Size	50 ants	100 chromosomes (GA), +50 ants (ACO)	30 particles (PSO) + 50 ants (ACO)
Iteration	50	200 generations (GA)	100(PSO)
Pheromone Evaporation Rate	0.5	0.5	0.5
Pheromone Influence(α)	1.0	1.0	1.0
Heuristic Influence(β)	2.0	2.0	2.0
Inertia Weight(w)	NA	NA	0.7 (reducing dynamically)
Cognitive Coefficient(ct)	NA	NA	1.5
Social Coefficient	NA	NA	1.5
Velocity Updates (PSO)	NA	NA	Based on inertia, cognitive and social components
Crossover Rate	NA	0.8(GA parameter)	NA
Mutation Rate (GA)	NA	0.05(GA parameter)	NA

To visualize the overall performance of these algorithms, graphs have been generated, illustrating their convergence behavior over time. Figure 2 provides a clean contrast of the routes built via every ACO version in terms of the exceptional performance of the answers.

Model Evaluation of ACO and Variants of ACO Algorithm

It demonstrates how the best route has been built using the different variants of the Ant Colony Optimization algorithm. Initially, the basic model for a route for the variant number of connected cities is from different ACO datasets. The corresponding visualization graphs are shown in Figure 2. Variants of the ACO algorithm have been compared in this examination, with every set of rules being evaluated primarily based on its capability to become aware of shorter and extra green routes.

Through iterative processes, each set of rules becomes adjusted to optimize its parameters, which include the pheromone replacement rate, evaporation price, and heuristic effect. On applying these procedures of ACO algorithms, the fitness of each algorithm is tuned optimally, as shown in Figure 3. Initially, route tracing was done locally by particle swarm optimization. This route is traced by applying the ant colony optimization, where the pheromone concentration is enhanced by setting the value as 1.3.

This enhancement caused improvements inside the neighborhood routes recognized by PSO, remodelling them into globally optimized answers. As a result, the blended PSO-ACO set of rules not only improved the efficiency of the quest procedure but also ensured that the routes identified were globally foremost. This improves the local route as best globally. Compared to the other two algorithms, PSO with ACO shows the best fitness.

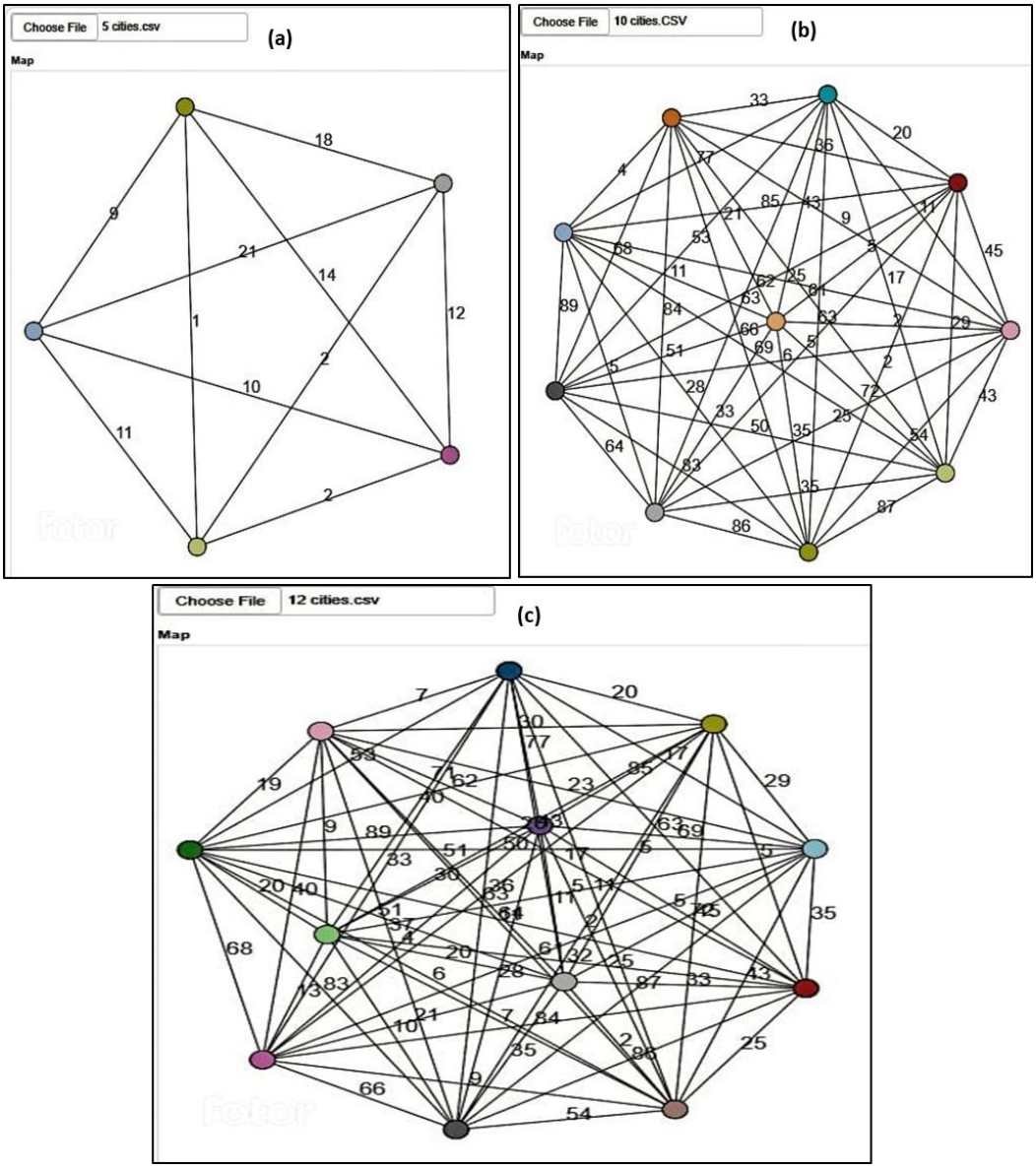
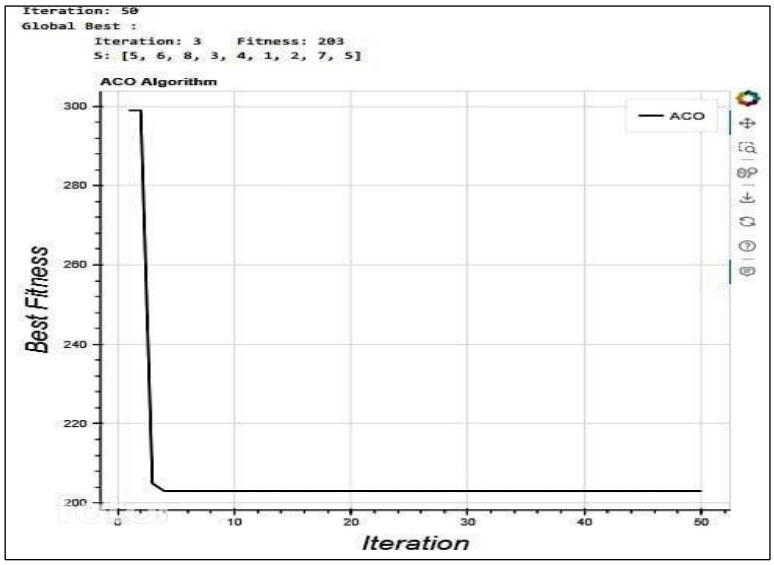
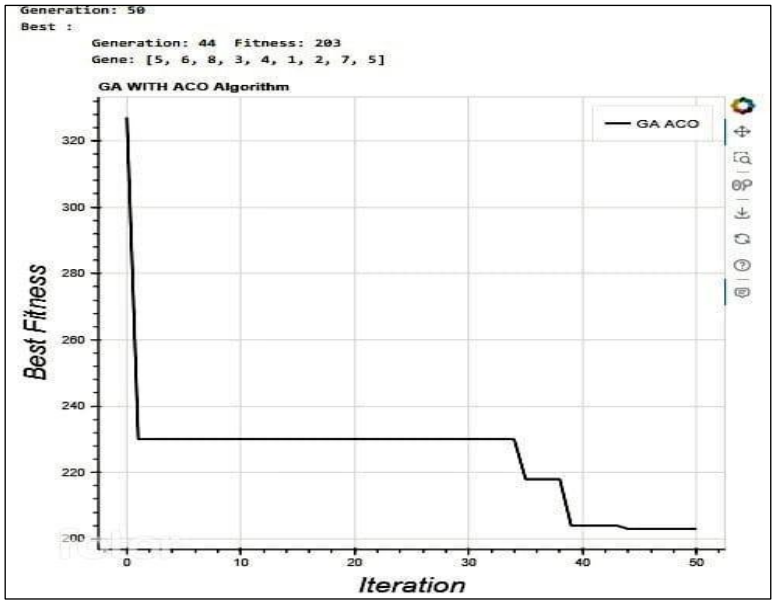


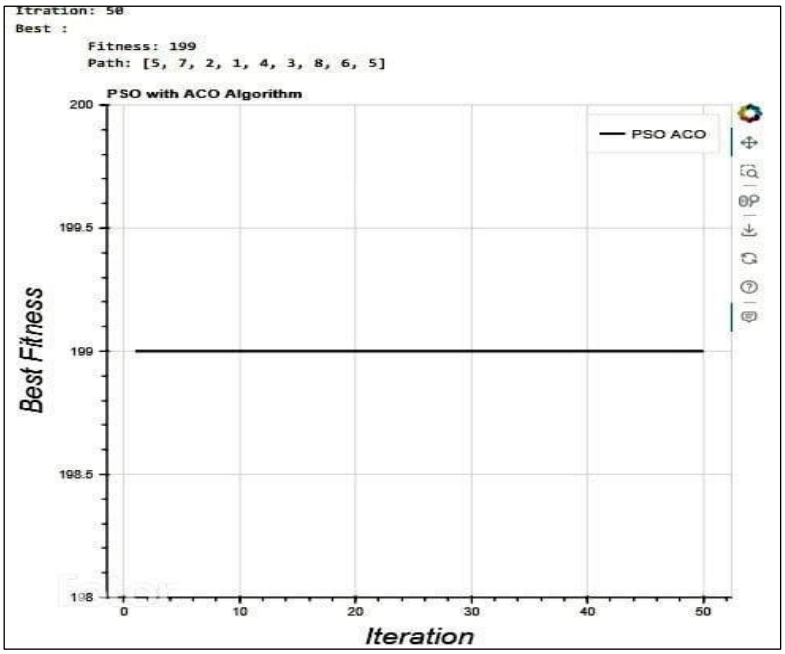
Figure 2: Dataset Visualization on Connected through the Following Number of (a) 5 (b) 10 (c) 12 Cities



(a) ACO Algorithm



(b) Enhanced GA - ACO Algorithm



(c) Enhanced PSO - ACO Algorithm

Figure 3: Best Fitness Obtained from ACO and Variants of ACO Algorithm

Performance Comparison for the Route Tracing Problem

Experiments have been conducted with dataset 4, which has 12 cities that produced 12 sets and implemented three optimization algorithms. The experiment settings ensure the effectiveness of the algorithms, and these details are accumulated in Table 3. Each experiment case constituted a different target city, which calculated the optimal path based on the estimated time duration.

Table 3 demonstrates that the optimal path for the dataset consisting of 12 cities has been estimated. The route distance of the ACO algorithm ranges from 163 to 226 cm, and the estimated speed range is from 4.01s to 4.98s. The optimal path has been estimated to be the route distance of the enhanced GA-ACO algorithm, which is from 164 to 216 cm, and the estimated speed range is from 3.24s to 4.07s. The optimal path has been estimated to be the route distance of the enhanced PSO-ACO algorithm, which is from 107 to 180 cm, and the estimated speed range is from 2.78s to 3.93s.

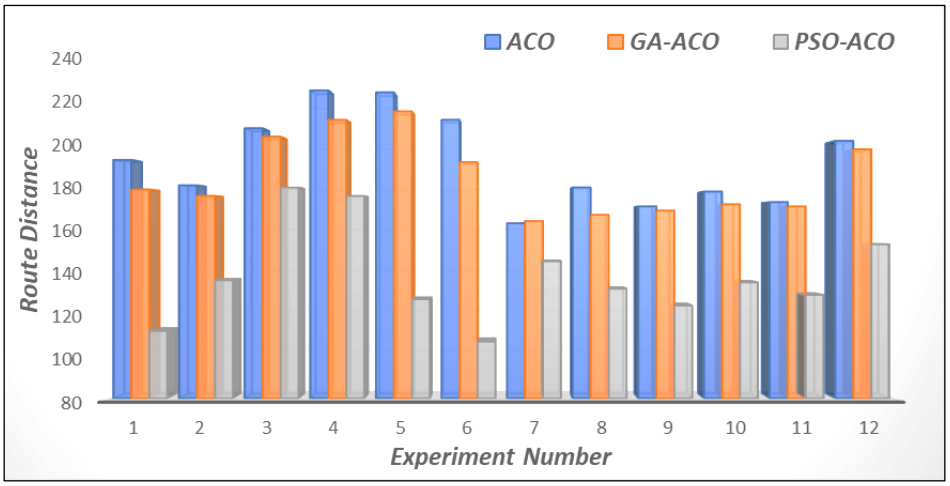
These experimental results are visualized and compared. The graphical points for 12 cases are shown in Figure 4. The enhanced PSO-ACO has a

better optimal path and is less time-consuming than the other two algorithms. Also, the enhanced GA-ACO algorithm has better results than the conventional ACO algorithm. Finally, the results of this experiment demonstrate that the enhanced PSO-ACO algorithm is more effective in route tracing and more significant than the enhanced GA-ACO in TSP applications.

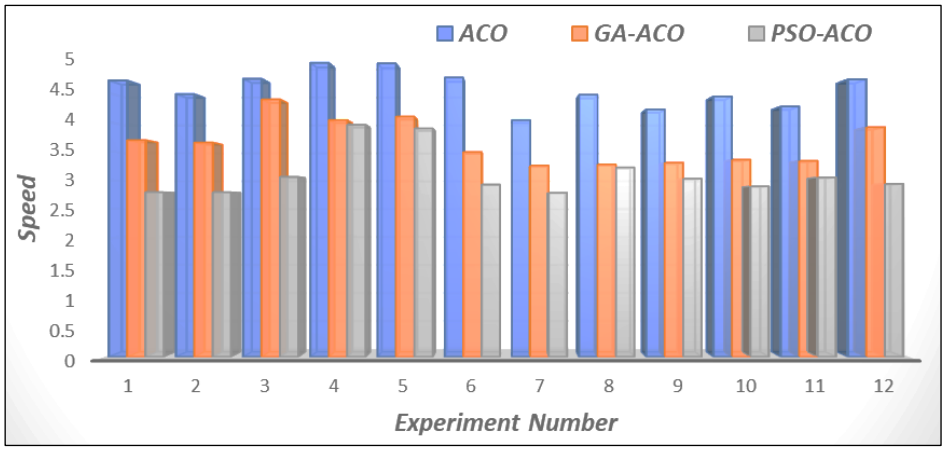
The optimization algorithms perform computations limited by 50 iterations, and their best cost is calculated by fitness applied to different datasets having varying numbers of cities. These values are listed in Table 4 and visually represented in Figure 5. Better results are obtained when enhanced ACO is integrated with the Genetic or Particle Swarm Optimization algorithm than the sole Ant Colony Optimization algorithm.

Table 3: Experimental Results of ACO and Variants of ACO Optimization Algorithms for the Dataset 4 (12 cities)

Ex. No.	ACO		GA-ACO		PSO-ACO	
	Route Distance d(cm)	Speed T(s)	Route Distance d(cm)	Speed T(s)	Route Distance d(cm)	Speed T(s)
1	193	4.68	179	3.67	112	2.79
2	181	4.45	176	3.63	136	3.05
3	208	4.71	204	4.36	180	3.93
4	226	4.98	212	4.01	176	3.87
5	225	4.97	216	4.07	127	2.92
6	212	4.73	192	3.47	107	2.78
7	163	4.01	164	3.24	145	3.21
8	180	4.44	167	3.26	132	3.02
9	171	4.19	169	3.29	124	2.89
10	178	4.41	172	3.34	135	3.04
11	173	4.24	171	3.32	129	2.93
12	202	4.70	198	3.89	153	3.27



(a) Experimental Results Based on Route Distance for Dataset of 12 Cities



(b) Experimental Results Based on Speed for Dataset of 12 Cities

Figure 4: Results of Experiments Cases Run on the ACO and Variants of ACO Algorithms Trained on a Dataset

Table 4: Performance Indicators of ACO and Variants of ACO Optimization Algorithms

Dataset with No. of Cities	ACO		GA - ACO		PSO – ACO	
	Speed	Fitness	Speed	Fitness	Speed	Fitness
Data Set 1 (5 Cities)	2.53	37	2.23	35	1.97	34
Data Set 2 (8 Cities)	3.09	204	2.98	203	2.09	199
Data Set 3 (10 Cities)	3.97	224	2.99	196	2.37	159
Data Set 4 (12 Cities)	4.77	216	3.43	190	2.71	103

The fitness function evaluates the corresponding algorithms to find the optimal path for the given TSP application. Table 4 determines that the best fitness leading to an optimal path of the ACO algorithm for various datasets, namely 1 to 4, is 37, 204, 224, and 216, respectively. The speed ranges from 2.53s to 4.77s. The best fitness leading to an optimal path of the GA-ACO algorithm for various datasets, namely 1 to 4, is 35, 203, 196, and 190, respectively. The speed ranges from 2.23s to 3.43s. The best fitness leading to an optimal path of the PSO-ACO algorithm for various datasets, namely 1 to 4, is 34, 199, 159, and 103, respectively. The speed ranges from 1.97s to 2.98s.

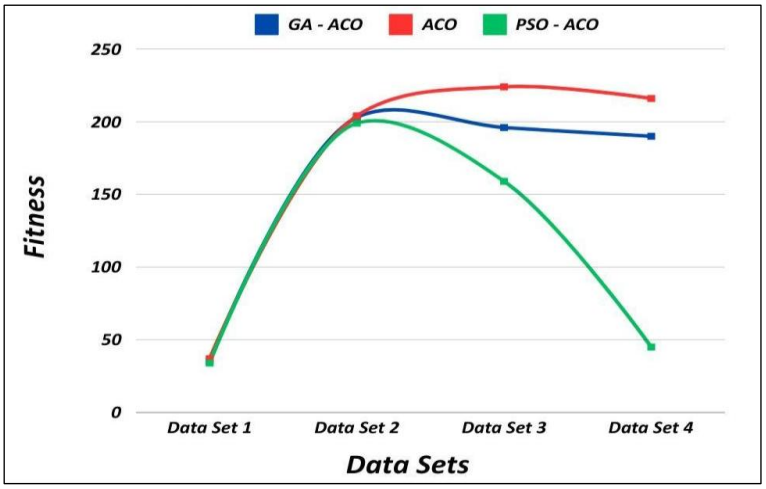


Figure 5: Performance Comparison of the ACO and Variants of ACO Algorithm Trained on Datasets

It is concluded that leveraging the strengths of PSO-ACO algorithms significantly changed the method of locating the most appropriate routes in complicated city networks. The software of ACO's pheromone-based seeks, in aggregate with the exploratory capabilities of PSO, to provide a powerful device for addressing huge-scale optimization problems. This research shows that the PSO-ACO hybrid method offers quality fitness results compared to different optimization algorithms, making it a perfect preference for solving complicated direction optimization issues, including those found inside the Traveling Salesman Problem.

Conclusion

This paper investigates solving the TSP using variants of Ant Colony Optimization (ACO), such as GA-ACO and PSO-ACO, that revealed nuanced performance differences among the algorithms. PSO-ACO consistently delivered high-quality solutions across various TSP instances, showcasing scalability and robustness, especially for larger problem sizes. GA-ACO demonstrated robustness in finding near-optimal solutions but with higher computational costs, while PSO exhibited fast convergence but struggled with global optimization. Each algorithm presented trade-offs between solution quality, convergence speed, and scalability, with ACO emerging as a versatile choice for tackling complex TSP instances. The user-friendly web application provided an intuitive platform for users to interact with and gain insights into these optimization techniques, emphasizing their efficacy in addressing real-world combinatorial optimization challenges.

Abbreviations

TSP: Travelling Salesman Problem
GA: Genetic Algorithm
PSO: Particle Swarm Optimization
ACO: Ant Colony Optimization

Acknowledgement

The authors expressed sincere gratitude to the organization for supporting the completion of this research work.

Author Contributions

S Selvi: Methodology and Script Preparation; D Dhinoovika: Conduct Experiments and Script Preparation; R Harshana and R Muthulakshmi: Experimental Analysis.

Conflict of Interest

The authors state that they have no regarded economic conflicts of interest or personal relationships that could have stimulated the work supplied in this paper.

Ethics Approval

Not applicable.

Funding

This research did not receive any funds.

References

1. Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation. 1997;1(1):53–66.
2. D'Acierno L, Gallo M, Montella B. An Ant Colony Optimization algorithm for solving the asymmetric traffic assignment problem. European Journal of Operational Research. 2012; 217(2):459–469.
3. Chang YC, Li VC, Chiang CJ. An ant colony optimization heuristic for an integrated production and distribution scheduling problem. Engineering Optimization. 2014;46(4):503-520.
4. Mohemmed AW, Sahoo NC, Geok TK. Solving shortest path problem using particle swarm optimization. Applied Soft Computing. 2008;8(4):1643-1653.
5. Yu M, Yue G, Lu Z, Pang X. Logistics terminal distribution mode and path optimization based on ant colony algorithm. Wireless Personal Communications. 2018;102:2969-2985. DOI:10.1007/s11277-018-5319-z.
6. Huang RH, Yang CL. Ant colony system for job shop scheduling with time windows. The International Journal of Advanced Manufacturing Technology. 2008;39(1):151-157.
7. Ah CW, Ramakrishna RS. A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE Transactions on Evolutionary Computation. 2022;6(6):566-579.
8. Abdallah H, Emara HM, Dorrah HT, Bahgat A. Using Ant Colony Optimization Algorithm for Solving Project Management Problems. Expert Systems with Applications. 2009;36(6):10004-10015.
9. Adamu, Patience I, Jegede, Joshua T, Okagbue, HI, Oguntunde PE. Shortest Path Planning Algorithm – A Particle Swarm Optimization (PSO) Approach. In: World Congress on Engineering. London, U.K. 2018 July 4-6.
10. Shaymaa Al-hayali, Osman N. Ucan, Oguz Bayat. Genetic Algorithm for finding shortest paths Problem. ICEMIS '18: Proceedings of the Fourth International Conference on Engineering & MIS. 2018;27:1–6.
11. Bell JE, McMullen PR. Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics. 2004;18(1):41-48.

12. McMullen PR. An ant colony optimization approach to addressing an IT sequencing problem with multiple objectives. Artificial Intelligence in Engineering. 2001;15(3):309-317.
13. Bergh van den F. An Analysis of Particle Swarm Optimizers. Thesis, University of Pretoria. 2002:15–30.
14. Scianna M. The Add ACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems. Mathematics and Computers in simulation. 2024;218:357-382. DOI:10.1016/j.matcom.2023.12.003.
15. Alipour MM, Razavi SN. A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem. Neural Computing and Applications. 2018;30:2935-2951.
16. Sedighizadeh D, Masehian E. Particle Swarm Optimization Methods, Taxonomy and Application. International Journal of Computer Theory and Engineering. 2009;1(5):1793-8201.
17. Gad AG. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Arch Computat Methods Eng. 2022;29:2531–2561. DOI:10.1007/s11831-021-09694-4.
18. Chen SM, Chien CY. Solving the Traveling Salesman Problem based on the Genetic Simulated Annealing Ant Colony System with Particle Swarm Optimization Technique. Expert Systems with Applications. 2011 Nov 1;38(12):14439-50.
19. Alatas B, Akin E. Chaotic Rough Particle Swam Optimization Algorithms. Eirat University Turkey. 2007. DOI: 10.5772/5094.
20. Murugananthan V, Rehan VMYES, Srinivasan R, Kavitha M, Kavitha R. Traveling Salesman Problem with Ant Colony Optimization. 2023 2nd International Conference on Edge Computing and Applications (ICECAA), India. 2023:481-485.
21. Song Q, Yu L, Li S, Hanajima N, Zhang X, Pu R. Energy Dispatching Based on an Improved PSO-ACO Algorithm. International Journal of Intelligent Systems. 2023;2023(1):3160184.