The Nepali Mathematical Sciences Report
Regd No.: 53/033-034, ISSN: 2392–411X
http://www.journal.cdmathtu.edu.np
Year: 2026, Volume: 43, No.: 1
Pages: 1–16
https://doi.org/10.3126/nmsr.v43i1.98493
Research Article

A Study on Solution Approaches to Travelling Salesperson Problem

*Prabin Gyawali[1] and Shree Ram Khadka[2]

[1,2]Central Department of Mathematics, Tribhuvan University, Kirtipur, Kathmandu, Nepal
prabin.gyawale@gmail.com, shreeramkhadka@gmail.com

Received: 21 March, 2026, Accepted: 5 July, 2026, Published Online: 10 August, 2026

Abstract

The study compares Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) for solving the symmetric Travelling Salesperson Problem (TSP) using real-world geographic data from heritage sites in the Kathmandu Valley.Using latitude–longitude data and distance matrices obtained from Google Maps, experiments were conducted on small (Ten), medium (Thirty), and large (Forty-Nine) size problem instances in Python.Performance is evaluated in terms of solution quality, convergence behavior, computational time, and scalability. The results indicate that all three algorithms perform comparably for small problem instances. As the problem size increases,Ant Colony Optimization consistently produces shorter tours and demonstrates more robust convergence and superior scalability.Ant Colony Optimization is identified as the most suitable approach for large, real-world symmetric TSP problems involving geographic data.

Keywords: Travelling Salesperson Problem, Ant Colony Optimization, Genetic Algorithm, Particle Swarm Optimization, Metaheuristic Optimization.
AMS(MOS) Subject Classification: 90C27, 90C59

1 Introduction

The Travelling Salesperson Problem is one of the most well-known and extensively studied combinatorial optimization problems in computer science, mathematics, and operations research[4, 7].Although its definition looks simple,the TSP requires rigorous computation and has become a standard benchmark problem due to its high computational complexity,

*Corresponding author

and practical applicability[15].The problem consists of determining a Hamiltonian tour that visits a given set of cities exactly once and returns to the starting city while minimizing the total travel cost or distance[12, 13].When distance is treated as cost,the TSP naturally models cost efficient routing problems [13].Beyond its theoretical significance, the Travelling salesperson problem has numerous real-world applications in transportation systems, logistics routing, tourism planning, and network design[15, 16].

To demonstrate it mathematically we can represent the Travelling salesperson problem using the graph $G = (V, E)$, where V denotes the set of vertices, that represents cities and E denotes to the part connecting them.A cost matrix $C = (c_{ij})$ is defined, where c_{ij} represents the cost like fare or opportunity cost,or distance of travelling from city i to city j. In many practical applications, including problems based on Euclidean distances, the triangle inequality

$$c_{ik} \leq c_{ij} + c_{jk} \tag{1.1}$$

is assumed to hold[3].

The TSP arises in different variants, among which the symmetric Travelling Salesperson Problem is the most commonly studied.In the Symmetric Travelling Salesperson (STSP) problem the travel cost between any two cities is identical in both directions, that is, $c_{ij} = c_{ji}$, resulting in an undirected graph representation[10].A symmetric version of TSP is used to simplify the comparison process since distances between different sites are assumed to be almost equal in both directions.In addition to symmetric versions, there are also other types of Travelling Salesperson Problem, such as asymmetric and multiple travelling salesperson problems[8].

Because the Travelling Salesman Problem is an NP-hard problem in which computational complexity grows exponentially with increasing size of the problem, and since there is a factorially growing number of potential routes in any instance, heuristics and meta-heuristics are popularly used to solve this problem by reaching optimal results.Exact methods such as branch-and-bound and cutting-plane approaches can solve small instances optimally but do not scale well[10].Consequently, heuristic and metaheuristic algorithms have been widely adopted to obtain high-quality approximate solutions within reasonable computational time.These include local search techniques such as 2 opt as well as population based metaheuristics such as genetic algorithms,ant colony optimization,and particle swarm optimization[10, 14, 16].These methods aim to balance exploration and exploitation of the search space and have proven effective for solving large symmetric TSP instances.

The TSP has a long research history and is one of the most studied and researched problems in combinatorial optimization.Problems closely related to the Travelling Salesperson problem can be traced back to Euler's work in 1759 on the knight's tour problem on a chessboard.The Travelling Salesperson problem itself was formally stated as a mathematical

optimization problem by Miller in 1930[6, 7].

Early research on the Travelling Salesperson Problem primarily focused on exact solution approaches.A significant breakthrough was achieved in 1954 when Johnson, Fulkerson, and Dantzig solved a 49-city Travelling Salesperson Problem instance.Following this, researchers progressively solved larger instances using exact methods. Held and Karp solved a 64-city instance in 1971, Camerini et al. solved 67 cities in 1975, and Grötschel solved 120 cities in 1977. During the 1980s, further progress was reported, including solutions for 318 cities by Crowder and Padberg in 1980, 532 cities by Padberg and Rinaldi in 1987, and 666 cities by Grötschel and Holland in the same year[7].

Advancements continued into the 1990s and early 2000s, significantly expanding the scale of solvable instances. Padberg and Rinaldi solved a 2392-city instance in 1991, while Applegate, Bixby, Chvátal, and Cook extended the solution size to 7397 cities in 1995. Subsequent milestones included solutions for 13,509 cities in 1998, 15,112 cities in 2001, and 24,978 cities in 2004 in collaboration with Helsgaun. Even larger instances were solved in later years, including 33,810 cities in 2007 by Cook, Espinoza, and Goycoolea, and 85,900 cities in 2009 by Applegate and his collaborators[7].

Despite these remarkable achievements, the computational cost of exact methods increases rapidly with problem size.This limits their applicability and usefulness toward large-scale and real-world TSP instances.This limitation motivated the development of heuristic and metaheuristic approaches that are efficient for finding high solution in relative computational time.

Among these approaches, Genetic Algorithms,introduced by John Holland in 1975[7], are inspired by the principles of natural evolution and operate on a population of candidate solutions using selection, crossover,and mutation operators. Ant Colony Optimization,first proposed by Marco Dorigo in 1992 and later extended in 1999[5, 8],is inspired by the foraging behavior of ants and relies on collective learning through pheromone reinforcement. Particle Swarm Optimization , introduced by Kennedy and Eberhart in 1995[7],is based on the social behavior of birds and fish,where candidate solutions move through the search space by learning from individual and global experiences.While GA emphasizes evolutionary adaptation,ant colony optimization exploits indirect communication among agents,and PSO relies on information sharing within a swarm to guide the search process.

Although ACO,GA and PSO have demonstrated strong results in solving the TSP, their comparative behavior in terms of solution quality, convergence characteristics, computational time, and scalability under consistent experimental settings remains an important area of investigation.Motivated by these considerations,this study focuses on a comparative evaluation of selected metaheuristic algorithms for solving symmetric Travelling Salesperson Problem instances.A comparison of the relative performances of modern metaheuristic algorithms shows that there is no one algorithm that is optimal for all TSP problems.It

underscores the necessity of further research and algorithm selection[1].The Traveling Salesman Problem has received extensive attention in the past, but existing studies have mostly considered artificial datasets for testing or evaluating the performance of particular algorithms.Comparison of algorithms based on actual geographic datasets, especially those related to Nepalese cultural tourism, is scarce.

2 Methods

Population-based metaheuristic algorithms are used for solving the symmetric Travelling Salesperson Problem (TSP),and they are particularly suitable for dealing with combinatorial optimization problems for which exact algorithms are computationally expensive.While heuristics yield solutions through problem-specific procedures,the metaheuristics are based on general search procedures,such as stochastic searches, which can help avoid getting trapped at local minima.These metaheuristic algorithms include ant colony optimization (ACO), genetic algorithm (GA), and particle swarm optimization (PSO).They are extensively used for solving STSP and are compared within a common experimental framework.The stochastic nature of the algorithms brings an element of randomness to the search process.

2.1 Ant Colony Optimization

In the Ant Colony Optimization a set of artificial ants in a incremental way construct feasible tours by moving probabilistically between cities. ACS attains discovery of new directions and knowledge exploitation, and both applies global and local pheromone update rules, and has been shown to outperform several standard optimizers such as simulated annealing and evolutionary computation[2].

Let $\tau_{ij}(t)$ be the pheromone's intensity related with edge (i, j) at iteration t, and let the heuristic information be defined as

$$\eta_{ij} = \frac{1}{d_{ij}}.$$

When a ant k is located at city i, the probability of selecting city j from the set of unvisited cities N_i^k is given by

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum\limits_{u \in N_i^k} [\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta}, & \text{if } j \in N_i^k, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where parameters α and β control the relative influence of pheromone intensity and heuristic information, respectively.

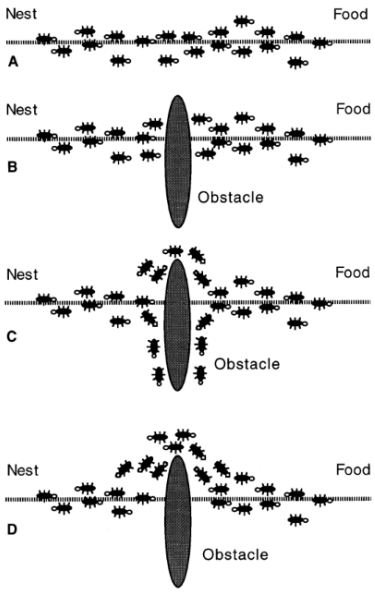

Figure 1: (A) Ant follow a path from nest to food, (B) obstacle appear in the path, (C) Pheromones are deposited in each side with equal probability, (D) ants deposit pheromones more quickly in the shorter path.[3]

Once all ants finish the tour, the process of updating the pheromone trail is done by evaporation and deposition. In particular, the evaporation process will regulate the decrease in pheromone strength over time in order to avoid too much pheromone build-up in chosen routes.:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \tag{2.2}$$

where $\rho \in (0, 1]$ is the evaporation rate, m is the number of ants, and

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, & \text{if ant } k \text{ traverses edge } (i, j), \\ 0, & \text{otherwise.} \end{cases}$$

Here, L_k denotes the total tour length constructed by ant k, and Q is a constant controlling pheromone deposition.

2.2 Genetic Algorithm

The Genetic Algorithm formulates the STSP as a population-based stochastic search process over the solution space.Among the fittest soltion, the values with higher fitness value are selected and combined in an attempt to produce better offspring. This gradual process, over many generations, improves the population to yield optimal or near-optimal

solutions[9].GA replicates the process by which nature evolves living things, but they operate computationally as a means to effectively solve real-world optimization problems with maximum efficiency[4, 9]. Let S be the set of all possible and feasible tours, and let each chromosome $x \in S$ represent a candidate solution. At generation t, the population is defined as

$$P(t) = \{x_1^{(t)}, x_2^{(t)}, \ldots, x_N^{(t)}\},$$

where N is the population size.

For a minimization objective function $J(x)$, the fitness of a chromosome is defined as

$$f(x) = \frac{1}{J(x) + \varepsilon}, \tag{2.3}$$

where $\varepsilon > 0$ is a small constant to ensure numerical stability.

Parent selection is performed probabilistically using roulette-wheel selection:

$$\Pr\{x_i^{(t)}\} = \frac{f(x_i^{(t)})}{\sum_{j=1}^{N} f(x_j^{(t)})}. \tag{2.4}$$

Given two parent chromosomes x_a and x_b, crossover is defined as

$$(x_a', x_b') = \begin{cases} C(x_a, x_b), & \text{with probability } p_c, \\ (x_a, x_b), & \text{with probability } 1 - p_c, \end{cases}$$

where p_c is the crossover probability. Mutation is applied independently to each chromosome:

$$x' = \begin{cases} M(x), & \text{with probability } p_m, \\ x, & \text{with probability } 1 - p_m, \end{cases}$$

where p_m is the mutation probability. The population is updated generationally as

$$P(t+1) = G(P(t); p_c, p_m). \tag{2.5}$$

2.3 Particle Swarm Optimization

In the Particle Swarm Optimization (PSO) each particle is a representational way of showcasing the possible and feasible solution to the STSP. Every particle is the candidate solution and have two features that are position illustrating the solution and velocity which covers direction and step length. PSO does not require gradient or derivative information, as many optimization techniques do. Because of this, it proves to be very effective for nonlinear, non-differentiable, and high-dimensional problems[11].

Let swarm consist of N particles. Every particle i is characterized by a position vector $x_i(t)$ and a velocity vector $v_i(t)$ at iteration t.

Every particle retains two best positions during the search process: $pbest_i$, the optimal position found particle which is the candidate solution i, and $gbest$, the best position found by the entire swarm.

The velocity and position are revised for every particle according to

$$v_i(t+1) = wv_i(t) + c_1 r_1(pbest_i - x_i(t)) + c_2 r_2(gbest - x_i(t)), \tag{2.6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{2.7}$$

where w is the inertia weight, c_1 and c_2 are cognitive, and social coefficients.

2.4 Stopping Criteria

All algorithms stop if at least one of following conditions is fulfilled:

- If maximum number of iterations is completed which is set beforehand,
- if there is no improvement in the best and optimal solution is observed over a defined number of iterations,
- Convergence behavior indicates stabilization of the solution.

These stopping conditions ensure an effective trade-off between solution quality and computational cost.

2.5 Performance Evaluation Criteria

To assess and compare the effectiveness of the proposed metaheuristic algorithms,a set of evaluation criteria is defined as part of the methodological framework.The performance of ACO, GA and PSO is evaluated based on the following measures:

- Solution quality: This criterion measures how close the obtained solution is to the best known or minimum tour length.It represents effectiveness of the algorithm in identifying high-quality routes.
- Convergence behavior: This metric evaluates how smoothly and efficiently an algorithm progresses toward a high-quality solution over successive iterations, indicating its search dynamics and stability.
- Computational time:This measure represents the total execution time required by an algorithm to reach a solution under the specified stopping criteria, providing insight into its computational efficiency.

- Scalability: This criterion examines how the algorithm's performance is affected as the problem size increases, with respect to the increase in the number of cities in the problem instance.

3 Problem Formulation

For each site considered in the study.Geographic coordinates were collected using Google Maps. Latitude and longitude values were used to represent the locations, providing a standardized and accurate description of real-world routing scenarios. Based on these coordinates, a symmetric dis- tance matrix was constructed by computing pairwise distances between all locations using a standard distance measure.

Kathmandu Valley is culturally important regions in the world, and contains seven UNESCO World Heritage Sites.Because of this it attracts a large number of domestic and international tourists each year.According to the 2024 UNESCO State of Conservation Report, most of the monuments damaged during the 2015 earthquake have been successfully restored.Despite, these restoration efforts the valley continues to face major challenges, in- cluding unplanned urban growth,traffic congestion,overcrowding at heritage sites,weak management practices and inadequate monitoring mechanisms.

In this context,the optimization of a heritage circuit,defined as identifying the most efficient,balanced and safe route connecting multiple heritage sites has emerged as an important research problem.Optimizing such routes can help reduce overcrowding, minimize travel time and cost for tourists, and it supports authorities in planning traffic movement and infrastructure de- velopment.A well-designed heritage circuit also contributes to the protection of culturally significant monuments by preventing excessive concentration of visitors, thereby supporting the long term preservation of Kathmandu Valley's cultural heritage.

Table 1: List of 49 Heritage and Cultural Sites with Coordinates

Code	Site Name	Latitude	Longitude	Remarks
S1	Pashupatinath Temple	27.70437344	85.30744154	
S2	Boudhanath Stupa	27.72152636	85.36208864	
S3	Swayambhunath (Monkey Temple)	27.71498506	85.29040642	
S4	Kathmandu Durbar Square	27.70437816	85.30745227	
S5	Narayanhiti Palace Museum	27.71482924	85.31804389	
S6	Garden of Dreams	27.71425246	85.31467908	
S7	Asan Bazaar	27.70729989	85.31180991	
S8	Thamel Chowk	27.71430216	85.31242162	
S9	Rani Pokhari	27.70758226	85.31490906	
S10	National Museum (Chhauni)	27.70586156	85.28951094	
S11	Kirtipur Historic Town	27.67477193	85.28045724	
S12	Kopan Monastery	27.74276599	85.36493448	
S13	Patan Durbar Square	27.67276601	85.32530899	
S14	Kumbheshwar Temple	27.67663159	85.32600756	
S15	Jawalakhel Zoo (Central Zoo)	27.67279899	85.31184012	
S16	Bungamati Heritage Village	27.61989292	85.30060502	
S17	Khokana Heritage Village	27.64308358	85.29519372	
S18	Ichangu Narayan Temple	27.73002389	85.26398712	
S19	Godawari Botanical Garden	27.59605357	85.38123677	
S20	Bhaktapur Durbar Square	27.67211480	85.42829021	
S21	Thimi Pottery Square	27.67938422	85.38485236	
S22	Changunarayan Temple	27.71643885	85.42775772	
S23	Siddha Pokhari	27.67191827	85.42097244	
S24	Nagarkot Viewpoint	27.71930619	85.50931909	
S25	Sanga Kailashnath Mahadev Statue	27.64625614	85.47448597	
S26	Chandragiri Hill	27.67827757	85.21125268	
S27	Dhulikhel Viewpoint	27.60920065	85.56550673	
S28	Phulchowki Hill	27.57087896	85.40664310	
S29	Sundarijal (Shivapuri Entry)	27.77236123	85.42555249	
S30	Dakshinkali Temple	27.60531614	85.26349135	
S31	White Monastery (Seto Gumba)	27.61355462	85.26031764	
S32	Manjushree Park (Chobhar)	27.66057555	85.29304836	
S33	Bajrabahari Temple	27.60638425	85.32926314	
S34	Guhyeshwari Temple	27.71126476	85.35330376	
S35	Kirateshwar Mahadev Temple	27.71252176	85.34983517	
S36	Rato Machhindranath Temple	27.67039641	85.32291045	
S37	Hiranya Varna Mahavihar (Golden Temple)	27.67526548	85.32446835	
S38	Krishna Mandir (Shikhara Style)	27.67367606	85.32487283	
S39	Nyatapola Temple	27.67140206	85.42933386	
S40	Dattatreya Temple	27.67350406	85.43541757	
S41	Pilot Baba Ashram	27.64017757	85.42274647	
S42	Panauti Heritage Town	27.58532573	85.51551192	
S43	Namobuddha Monastery	27.57136682	85.58251647	
S44	Shivapuri Peak	27.81230924	85.38456209	
S45	Gokarna Mahadev Temple	27.73947949	85.38767274	
S46	Sankhu Ancient Town	27.73044710	85.46451264	
S47	Vajrayogini Temple (Sankhu)	27.74361769	85.46721564	
S48	Sanga Mahadev–View Tower Area	27.64612947	85.47419483	
S49	Taudaha Lake	27.64911733	85.28174243	

3.1 Problem 1: Small-Scale TSP (10 Sites)

This problem consists of 10 major heritage and cultural sites in the Kathmandu Valley.It is a simple and computationally light instance designed to test the basic working behaviour of the algorithms.

Table 2: Distance Matrix for 10-Site TSP Instance (in km)

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	0.00	2.21	9.21	6.08	5.73	17.21	17.99	6.27	7.01	13.93
S2	2.21	0.00	11.81	9.02	7.45	22.15	21.66	6.65	5.52	12.66
S3	9.21	11.81	0.00	3.51	10.73	12.67	10.63	14.71	15.89	8.19
S4	6.08	9.02	3.51	0.00	7.64	12.21	11.91	11.97	12.61	10.60
S5	5.73	7.45	10.73	7.64	0.00	11.83	18.04	6.94	11.37	15.98
S6	17.21	22.15	12.67	12.21	11.83	0.00	4.08	18.82	21.64	19.89
S7	17.99	21.66	10.63	11.91	18.04	4.08	0.00	22.70	25.63	17.84
S8	6.27	6.65	14.71	11.97	6.94	18.82	22.70	0.00	9.15	19.06
S9	7.01	5.52	15.89	12.61	11.37	21.64	25.63	9.15	0.00	17.38
S10	13.93	12.66	8.19	10.60	15.98	19.89	17.84	19.06	17.38	0.00

3.2 Problem 2: Medium-Scale TSP (30 Sites)

The medium-scale problem includes 30 culturally and historically signifi- cant locations across the Kathmandu Valley. This instance introduces mod- erate complexity for comparing solution quality and convergence behaviour. Since the complete 30×30 matrix is too large to include in the journal paper,a portion of the matrix is shown below for illustration. The full matrix is available with authors

$$D = \begin{matrix} 0.00 & 2.21 & 8.80 & 6.08 & 5.51 & \cdots \\ 2.21 & 0.00 & 11.81 & 7.54 & 10.21 & \cdots \\ 8.80 & 11.81 & 0.00 & 3.51 & 9.26 & \cdots \\ 6.08 & 7.54 & 3.51 & 0.00 & 7.64 & \cdots \\ 5.51 & 10.21 & 9.26 & 7.64 & 0.00 & \cdots \\ & \cdots & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

3.3 Problem 3: Large-Scale TSP (49 Sites)

This large-scale problem includes all 49 important destinations from all across the Kathmandu Valley. It is designed to evaluate the scalability, robustness,and computational

efficiency of ACO, GA, and PSO in a high dimensional search space.Since the complete 49 × 49 matrix is too large to include in the journal paper, a portion of the matrix is shown below for illustration.The full matrix is available with authors.

$$D = \begin{matrix} 0.00 & 2.21 & 8.80 & 6.08 & 5.51 & \cdots \\ 2.21 & 0.00 & 11.81 & 7.54 & 10.21 & \cdots \\ 8.80 & 11.81 & 0.00 & 3.51 & 9.26 & \cdots \\ 6.08 & 7.54 & 3.51 & 0.00 & 7.64 & \cdots \\ 5.51 & 10.21 & 9.26 & 7.64 & 0.00 & \cdots \\ & \cdots & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

4 Results and Discussion

This section presents a comparative analysis of ACO , GA , and PSO applied to the Travelling Salesperson Problem using real-world heritage and cultural locations of the Kathmandu Valley.The evaluation is performed on three different problem sizes (small 10, medium 30, and large 49 sites) and the results are measured in terms of the quality of the solution, convergence characteristics, computation time, and scalability; importantly, the solutions that were obtained are close to optimum but not global optimum as the case may be with meta-heuristics.

4.1 Solution Quality

Solution quality is evaluated based on the total length of the final tour produced by each algorithm, where shorter tour lengths indicate better so- lutions. latitude and longitude are used to locate the sites.

For the small-scale instance with 10 sites, all three algorithms achieved near-optimal solutions, with minimal differences in tour length due to the limited search space.As the problem size increased to 30 sites, differences in solution quality became more evident.Ant Colony Optimization and Parti cle Swarm optimization consistently produced shorter routes than Genetics Algorithm, indicating more effective search-space exploitation. Ant Colony Optimization benefited from pheromone reinforcement, while PSO achieved competitive solutions through rapid early convergence. In the large-scale instance with 49 sites, Ant Colony Optimization con sistently produced the shortest tour length, demonstrating superior solution quality. GA generated acceptable but longer routes, while Particle Swarm Optimization exhibited a noticeable decline in solution quality, largely due to premature convergence. These results indicate that ACO maintains higher solution quality as problem dimensionality increases

4.2 Convergence Behavior

Convergence behavior reflects how efficiently an algorithm approaches a high-quality solution over successive iterations. For the 10-site instance, all algorithms converged rapidly. PSO demonstrated the fastest convergence, while ACO showed minor early fluctuations due to exploratory behavior. GA required more generations to converge, reflecting the overhead of evolutionary operators.

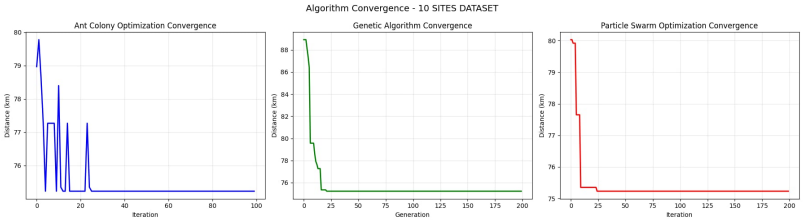

Figure 2: Convergence behavior of ACO, GA, and PSO for the 10-site dataset

For the 30-site instance, ACO exhibited fluctuating convergence in early and middle iterations, allowing it to escape suboptimal solutions and gradually refine results. GA showed smoother but slower convergence, while PSO improved rapidly in early iterations but stagnated later.

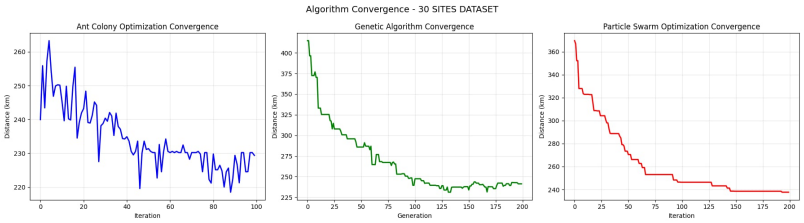

Figure 3: Convergence behavior of ACO, GA, and PSO for the 30-site dataset

In the 49-site instance, ACO demonstrated the most robust convergence behavior, continuing to improve solution quality even in later iterations. GA showed signs of premature convergence, and PSO stagnated early, indicating limited exploration in the high-dimensional search space.

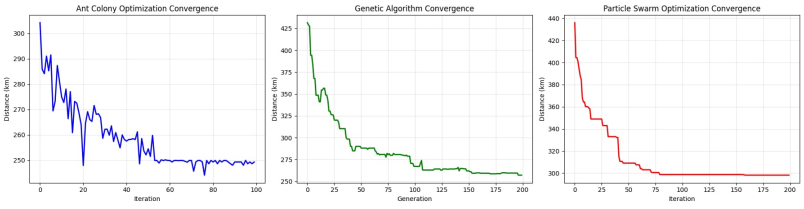

Figure 4: Convergence behavior of ACO, GA, and PSO for the 49-site dataset

4.3 Computation Time

Computation time is influenced by problem size and algorithmic complexity. PSO required the least computation time due to its simple update rules, making it efficient for small and medium-scale problems. GA exhibited moderate computation time, balancing robustness and computational cost through evolutionary operators. ACO required the highest computation time, as multiple ants construct full tours and pheromone updates are applied to all edges. However, this additional computational effort resulted in superior solution quality, particularly for large-scale instances.

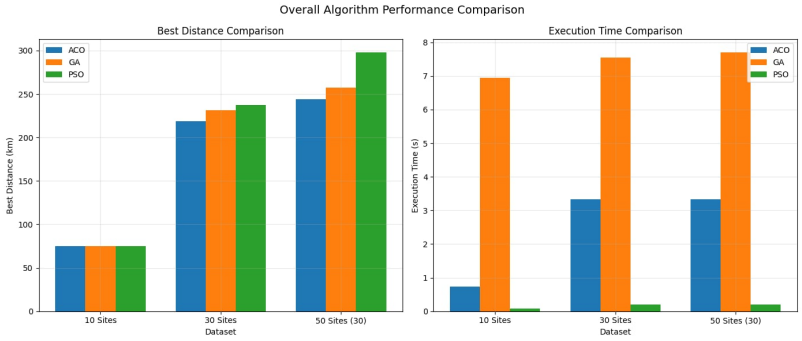

Figure 5: Histogram with illustrating execution and performance

4.4 Scalability

Scalability is assessed by analyzing algorithm performance as the number of sites increases from 10 to 49.While all algorithms performed similarly for small instances, performance differences became more pronounced as problem size increased.ACO scaled most effectively, maintaining high solution quality and continued improvement for larger instances.The consistent performance of ACO across different problem sizes demonstrates its robustness, scalability, and potential applicability to similar real-world routing and tourism optimization problems.While GA showed reasonable scalability and consistent performance, it exhibited premature convergence along with high computational cost as the problem became larger, with solution quality relying heavily on the parameters used and crossover technique employed.PSO exhibited limited scalability, as early stagnation reduced its effectiveness in larger search spaces.

4.5 Summary of Findings

The comparative analysis demonstrates that all three algorithms are effective for small-scale TSP instances.For medium-scale problems,ACO and PSO showed stronger performance than GA, with ACO providing more consistent improvement.In large-scale TSP instances,

ACO clearly outperformed GA and PSO with reference of solution quality, convergence robustness, and scalability.Ant Colony Optimization comes as the most robust and scalable algorithm for large, real-world TSP instances involving geographic data.GA provided reliable and stable performance with moderate computational cost, while PSO proved most suitable for small to medium-scale problems where rapid convergence is prioritized.

5 Conclusion

After a comparative analysis of Ant Colony Optimization , Genetic Al- gorithm and Particle Swarm Optimization for solving the Traveling Salesman Problem using real-world geographic data from heritage sites in the Kathmandu Valley. The experimental results show that all three algorithms perform comparably for small-scale problem instances.However, as the problem size increases, Ant Colony Optimization consistently produces higher-quality solutions and demonstrates superior scalability.Genetic algorithm exhibits stable but slower convergence, while PSO achieves faster execution at the cost of reduced performance in large-scale instances.

Ant Colony Optimization emerges as the most effective approach for solving large and complex TSP problems, whereas Genetic algorithm and Ant colony Optimization remain suitable for smaller in- stances depending on computational requirements.Weaknesses associated with Genetic Algorithms can be minimized using hybrid optimization techniques and efficient search methods for near-optimal solutions.Though this research is concerned with symmetric cases, the meta-heuristic techniques developed herein may also be extended to asymmetrical routing problems with certain changes in the modeling of the cost distances between cities.

6 Limitations

The primary objective of this study involves comparing ACO, GA, and PSO algorithms within a similar experimental setup using geographical distances from the Kathmandu Valley area. While the study takes into account practical applications of tourist route planning, the distance matrix is symmetric because only the cost associated with traveling distance is taken into consideration, while other factors, including fuel costs, budget considerations, traffic conditions, and time window constraints, among others, are ignored. Another limitation of the study includes the fact that the study uses a small-scale database. The scope of future studies could be expanded on large-scale and complex databases.

Acknowledgment

This research was financially supported by the University Grants Commission(UGC) through the award of the M.Phil. Fellowship (M.Phil-81/82)

References

[1] S.M. Almufti, A.A. Shaban, Comparative analysis of metaheuristic algorithms for solving the travelling salesman problems, International Journal of Scientific World, Vol. 11(2), pp 26-30, 2025.

[2] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, Theoretical Computer Science, Vol. 344, pp 243-278, 2005.

[3] M. Dorigo, L.M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, Vol. 1, pp 53-66, 1997.

[4] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, BioSystems, Vol. 43, pp 73-81, 1997.

[5] D. Gaertner, K. Clark, On optimal parameters for ant colony optimization algorithms, Proceedings of IC-AI, pp 83-89, 2005.

[6] S. Goyal, A survey on travelling salesman problem, Proceedings of the Midwest Instruction and Computing Symposium, pp 1-9, 2010.

[7] U. Klansek, Using the TSP solution for optimal route scheduling in construction management, Organization, Technology and Management in Construction, Vol. 3, pp 243-249, 2011.

[8] N. Kumbharana, G.M. Pandey, A comparative study of ACO, GA and SA for solving travelling salesman problem, International Journal of Societal Applications of Computer Science, Vol. 2, pp 224-228, 2013.

[9] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognition, Vol. 33, pp 1455-1465, 2000.

[10] C. Moon, J. Kim, G. Choi, Y. Seo, An efficient genetic algorithm for the traveling salesman problem with precedence constraints, European Journal of Operational Research, Vol. 140, pp 606-617, 2002.

[11] M.A. Qasimi, Solving the travelling salesman problem using the PSO optimization, TechComp Innovations Journal of Computer Science and Technology, Vol. 1, pp 19-27, 2024.

[12] H. Rania, C. Babak, D.W. Olivier, A comparison of particle swarm optimization and the genetic algorithm, American Institute of Aeronautics and Astronautics, Vol. 1, pp 1-13, 2005.

[13] C. Rego, D. Gamboa, F. Glover, C. Osterman, Traveling salesman problem heuristics: Leading methods, implementations and latest advances, European Journal of Operational Research, Vol. 211, pp 427-441, 2011.

[14] N. Rokbani, A. Abraham, A. Haqiq, Solving the travelling salesman problem using fuzzy and simplified variants of ant supervised by PSO with local search policy, International Journal of Hybrid Intelligent Systems, Vol. 15, pp 1-10, 2018.

[15] X. Yan, C. Zhang, W. Luo, W. Li, W. Chen, H. Liu, Solve traveling salesman problem using particle swarm optimization algorithm, International Journal of Computer Science Issues, Vol. 9, pp 264-271, 2012.

[16] 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, pp 979-987, 2019.