Comparative Analysis of Metaheuristic Algorithms for Tourism Route Optimization Using Real Road Network Distances: A Case Study of Yogyakarta, Indonesia

First Author[1*], Second Author[2], Third Author[2]
[1]Department, University, City, Indonesia
email@university.ac.id
[2]Department, University, City, Indonesia
[*]Corresponding author

***Abstract*—This study evaluates four metaheuristic algorithms for tourism route optimization in Yogyakarta Special Region, Indonesia, formulated as a Travelling Salesman Problem over 25 tourist attractions. Road distances were computed via Dijkstra's algorithm on an OpenStreetMap network of 153,334 nodes and 200,104 edges. Simulated Annealing (SA) with 2-opt produced the shortest mean route of $283.82\,\mathrm{km}$ ($\sigma = 0.08\,\mathrm{km}$), followed by Max–Min Ant System (MMAS, $284.35\,\mathrm{km}$), Ant Colony System (ACS, $285.49\,\mathrm{km}$), and Genetic Algorithm (GA, $302.13\,\mathrm{km}$). All pairwise differences were significant by Wilcoxon signed-rank tests ($p < 0.05$). ACS was 10.7 times faster than SA. Parameter sensitivity analysis of 27 MMAS configurations revealed strong interactions between the pheromone weight α and evaporation rate ρ. A mean road-to-Euclidean ratio of 1.18 confirms that real road data is necessary for practical route planning.**

***Index Terms*—tourism route optimization, Travelling Salesman Problem, Ant Colony Optimization, Simulated Annealing, Genetic Algorithm, OpenStreetMap, Yogyakarta**

I. Introduction

Yogyakarta Special Region (Daerah Istimewa Yogyakarta, DIY) receives more than 4.5 million visitors annually, offering attractions from the UNESCO-listed Borobudur and Prambanan temples to coastal and volcanic landscapes [1]. Visitors wanting to cover many sites in limited time face a route-planning problem that maps onto the Travelling Salesman Problem (TSP), which is NP-hard and grows factorially with the number of locations [2], [3].

Metaheuristic algorithms provide approximate solutions within practical time budgets. Ant Colony Optimization (ACO) variants exploit pheromone-based reinforcement [4], [5], Genetic Algorithms (GA) use selection, crossover, and mutation [6], and Simulated Annealing (SA) escapes local optima through a cooling schedule [7]. Their relative performance depends on problem size, distance metric, and parameter settings [2], [3].

A persistent shortcoming in tourism route studies is the reliance on Euclidean or haversine distances [8]. Road networks contain one-way streets, bridges, and topographic detours causing actual distances to exceed straight-line estimates by 15–20%. Yogyakarta's volcanic-slope-to-coast geography amplifies these discrepancies.

This paper addresses this gap by computing a 25×25 road-distance matrix via Dijkstra's algorithm on the OpenStreetMap network for DIY (153,334 nodes, 200,104 edges) and comparing Max–Min Ant System (MMAS), Ant Colony System (ACS), GA, and SA over 30 independent runs. The comparison covers solution quality, variance, computation time, convergence behaviour, and statistical significance. Parameter sensitivity and scalability analyses complement the main experiment.

II. Related Work

A. TSP and Metaheuristics

The TSP seeks the minimum-cost Hamiltonian cycle through n cities; for symmetric TSP the solution space contains $(n-1)!/2$ tours [2]. Advanced heuristics such as LKH [9] produce near-optimal solutions for large instances, but population-based and trajectory-based metaheuristics remain the practical choice for resource-constrained settings [10].

Dorigo et al. [4] introduced Ant System; Stützle and Hoos [5] proposed MMAS with bounded pheromone trails; Dorigo and Gambardella [11] developed ACS with pseudo-random selection and local pheromone decay. Recent ACO work includes adaptive heuristics [12], large-scale strategies [13], and hybrid methods [14], [15]. GA for TSP uses permutation-preserving operators such as OX crossover [6], [16], sometimes combined with reinforcement learning [17]. SA with 2-opt [7], [18] remains competitive for moderate instances [19].

B. Tourism Route Optimization

Ruiz-Meza and Montoya-Torres [20] surveyed tourist trip design, covering orienteering formulations with time windows [21], [22]. Sun et al. [23] proposed multi-objective ACO for route recommendation. Most studies use Euclidean distances; OSMnx [24], [25] has made real road extraction practical, and Boyaci et al. [8] and Tatit et al. [26] confirmed that Euclidean approximations introduce systematic errors. Within

Indonesia, Nasution et al. [27] applied vehicle routing to tourism itineraries, and Fathurrohman et al. [28] formulated a green orienteering problem for Yogyakarta [29]. Neither provided the multi-algorithm statistical comparison over real road distances offered here.

III. Materials and Method

A. Problem Formulation

Let $V = \{1, 2, \ldots, n\}$ be $n = 25$ tourist attractions. The objective is to find a permutation π minimising:

$$D(\pi) = \sum_{i=1}^{n-1} d(\pi_i, \pi_{i+1}) + d(\pi_n, \pi_1) \tag{1}$$

where $d(i, j)$ is the shortest road distance between attractions i and j.

B. Study Area and Data

Twenty-five attractions in DIY were selected across six categories. Table I lists each attraction with its geographic coordinates. The road network was downloaded via OSMnx [24] using the query "Daerah Istimewa Yogyakarta, Indonesia" with `network_type=drive`, yielding 153,334 nodes and 200,104 edges after undirected conversion. Each attraction was snapped to its nearest node, and the 25×25 distance matrix was computed using Dijkstra's algorithm. Fig. 1 illustrates the spatial distribution of these 25 attractions across the study area; the spread from urban Yogyakarta city to remote coastal and highland sites is clearly visible in the map.

C. Algorithm Implementations

All algorithms were implemented in Python with NumPy vectorisation. The core mechanisms are formalised below.

ACO transition rule. Both MMAS and ACS construct tours by having $m = 25$ ants choose the next city probabilistically. For ant k at city i, the probability of moving to unvisited city j is:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \, [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha \, [\eta_{il}]^\beta}, \quad j \in \mathcal{N}_i^k \tag{2}$$

where τ_{ij} is the pheromone on edge (i, j), $\eta_{ij} = 1/d(i, j)$ is the heuristic desirability, $\mathcal{N}_i^k$ is the set of unvisited cities, and α, β control the relative influence of pheromone versus distance.

MMAS [5] restricts pheromone trails to $[\tau_{\min}, \tau_{\max}]$. After each iteration, only the best ant deposits pheromone:

$$\tau_{ij} \leftarrow \text{clamp}\big[(1 - \rho)\, \tau_{ij} + \Delta\tau_{ij}^{\text{best}},\ \tau_{\min},\ \tau_{\max}\big] \tag{3}$$

where $\Delta\tau_{ij}^{\text{best}} = 1/D^{\text{best}}$ if edge (i, j) belongs to the best tour. Bounds are set as $\tau_{\max} = 1/(\rho \cdot D_{\text{NN}})$ and $\tau_{\min} = \tau_{\max}/(2n)$. Parameters: $\alpha = 1.0$, $\beta = 3.0$, $\rho = 0.02$, 500 iterations, stagnation re-initialisation after 100 iterations without improvement.

ACS [11] uses a pseudo-random proportional rule: with probability q_0 the ant greedily selects $\arg\max_j [\tau_{ij}]^\alpha [\eta_{ij}]^\beta$; otherwise it samples from Eq. (2). After each step, local pheromone decay is applied: $\tau_{ij} \leftarrow (1 - \xi)\, \tau_{ij} + \xi\, \tau_0$, with $\tau_0 = 1/(n \cdot D_{\text{NN}})$. Parameters: $q_0 = 0.9$, $\xi = 0.1$, $\rho = 0.1$, 500 iterations.

GA used Order Crossover (OX) on permutation chromosomes with population 100, crossover rate 0.8, swap mutation rate 0.02, tournament selection ($k = 3$), 10% elitism, 500 generations with early stopping after 100 generations without improvement.

SA explores the neighbourhood by 2-opt moves. At temperature T, a candidate solution with cost difference $\Delta D = D' - D$ is accepted with probability:

$$P(\text{accept}) = \begin{cases} 1 & \text{if } \Delta D < 0 \\ \exp(-\Delta D / T) & \text{otherwise} \end{cases} \tag{4}$$

A 2-opt move reverses a subtour segment between positions i and j. The cost change is evaluated in $O(1)$:

$$\Delta D = d(\pi_i, \pi_j) + d(\pi_{i+1}, \pi_{j+1}) - d(\pi_i, \pi_{i+1}) - d(\pi_j, \pi_{j+1}) \tag{5}$$

Parameters: $T_0 = 10{,}000$, cooling rate $\gamma = 0.999$, $T_{\text{end}} = 1.0$, 50 iterations per temperature step ($\sim$9,200 total steps).

Nearest Neighbour (NN) baseline was executed from all 25 starting nodes; the shortest tour was reported.

D. Experimental Design

Each stochastic algorithm was executed 30 times with seeds 42–71. The experimental programme consisted of four parts: (1) main comparison, with 120 stochastic runs (30 per algorithm) plus one deterministic NN run; (2) parameter sensitivity, testing 27 MMAS configurations by combining three levels each of pheromone weight (0.5, 1.0, 2.0), heuristic weight (2, 3, 5), and evaporation rate (0.02, 0.05, 0.10), with 10 runs per configuration; (3) scalability analysis on subsets of 10, 12, 15, and 20 attractions, 10 runs each; and (4) Wilcoxon signed-rank tests at a 0.05 significance level for all six pairwise algorithm comparisons.

Performance metrics. Algorithm quality is measured by the Relative Percentage Deviation (RPD) from the best-known solution D^*:

$$\text{RPD} = \frac{D - D^*}{D^*} \times 100\% \tag{6}$$

Solution consistency is assessed by the Coefficient of Variation (CV), defined as the standard deviation divided by the mean distance. Computational efficiency is reported as mean wall-clock time per run.

IV. Results and Discussion

A. Algorithm Performance

Table II summarises the results of all 30 independent runs for each algorithm. As the table shows, SA and MMAS both discovered the same best-known tour of $283.76\,\text{km}$, but SA was far more consistent: its mean distance was only $0.06\,\text{km}$ above the best, with a standard deviation of $0.08\,\text{km}$. The RPD and CV columns in Table II quantify this gap: SA's RPD from the optimum was just 0.02% with a CV of 0.03%, while

TABLE I: Tourist attractions in Yogyakarta Special Region ($n = 25$).

ID	Name	Lat.	Long.	ID	Name	Lat.	Long.
1	Kraton Yogyakarta	−7.805	110.364	14	Monumen Jogja Kembali	−7.750	110.377
2	Taman Sari Water Castle	−7.810	110.359	15	Purawisata	−7.801	110.374
3	Benteng Vredeburg Museum	−7.800	110.366	16	Candi Prambanan	−7.752	110.491
4	Tugu Yogyakarta	−7.783	110.367	17	Candi Ratu Boko	−7.770	110.489
5	Malioboro Street	−7.793	110.366	18	Tebing Breksi	−7.762	110.504
6	Pasar Beringharjo	−7.798	110.366	19	Museum Ullen Sentalu	−7.604	110.426
7	Museum Sonobudoyo	−7.802	110.364	20	Candi Borobudur	−7.608	110.204
8	Alun-Alun Kidul	−7.812	110.364	21	Pantai Parangtritis	−8.025	110.326
9	Alun-Alun Utara	−7.803	110.364	22	Hutan Pinus Mangunan	−7.931	110.431
10	Taman Pintar Science Park	−7.801	110.367	23	Goa Pindul	−7.953	110.650
11	Kebun Binatang Gembira Loka	−7.806	110.395	24	Pantai Indrayanti	−8.150	110.613
12	Museum Affandi	−7.783	110.397	25	HeHa Sky View	−7.855	110.454
13	Kotagede Heritage Area	−7.827	110.398				

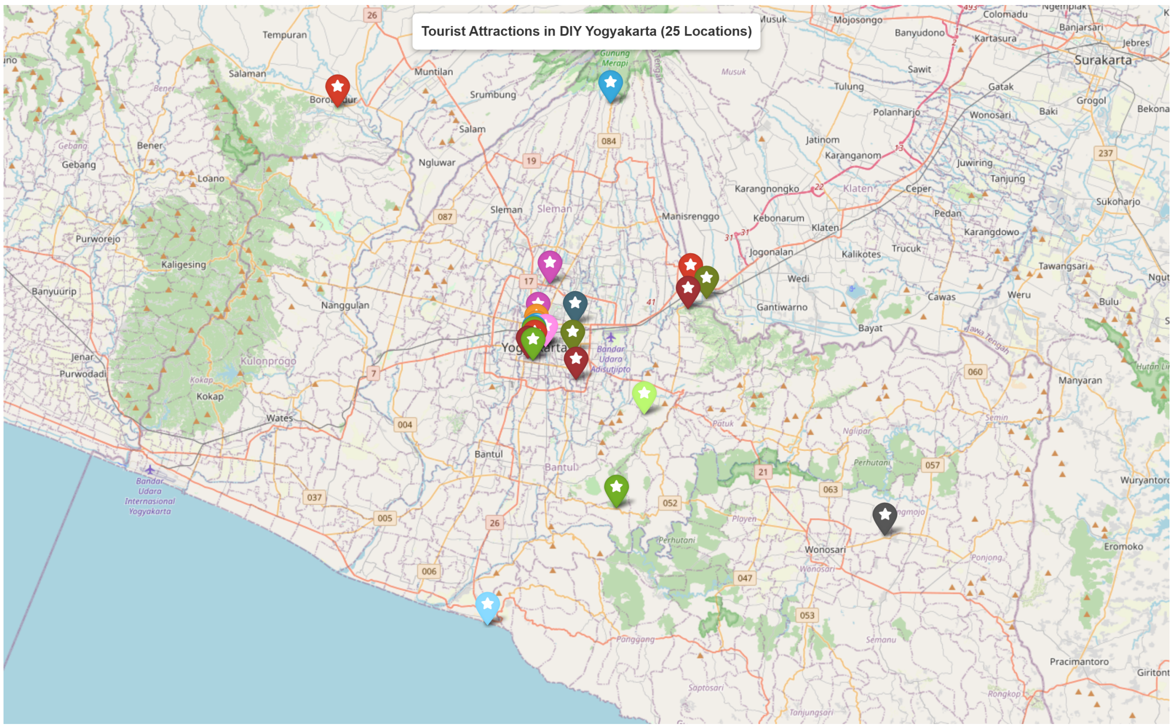


Fig. 1: Spatial distribution of 25 tourist attractions in Yogyakarta Special Region.

GA's RPD reached 6.47% with a CV of 2.25%. The trade-off between quality and speed is also visible in the table: although ACS tours averaged about $1.7\,\mathrm{km}$ longer than SA's, ACS completed each run 10.7 times faster ($0.327\,\mathrm{s}$ versus $3.496\,\mathrm{s}$). GA was the weakest performer overall; its mean tour length exceeded even the deterministic NN baseline, suggesting that basic OX crossover alone is insufficient for this problem size.

Fig. 2 illustrates the run-level distributions for each algorithm. As the box plots and violin overlays show, SA found the global optimum in 19 of its 30 runs; the remaining runs converged to one of three near-optimal tours, all within $0.21\,\mathrm{km}$ of the best. This tight clustering in Fig. 2 confirms that SA's 2-opt neighbourhood search consistently funnels solutions toward a small set of structurally similar tours. MMAS reached the optimum only once, but 19 of its 30 runs fell within 0.06% of the best-known distance. The violin panel of Fig. 2 reveals a bimodal distribution for ACS: roughly half the runs converged to its best tour, while the other half settled on longer routes. This split arises from the high exploitation probability (0.9), which can lock the colony onto established pheromone trails before sufficient exploration has occurred. GA produced the widest spread, with a $26.7\,\mathrm{km}$ gap between its best and worst runs, and no concentration around any particular tour structure.

To verify whether the observed differences are statistically meaningful, Table III reports the Wilcoxon signed-rank test results for all six pairwise comparisons. As Table III shows, every pair is statistically significant ($p < 0.05$), establishing the ranking SA $>$ MMAS $>$ ACS $>$ GA. Even the closest pair, MMAS versus ACS, yielded $p = 0.011$, while the remaining five comparisons all returned $p < 0.0001$. For every pair involving GA, all 30 matched observations favoured the competing algorithm ($W = 0$).

TABLE II: Algorithm performance comparison over 30 independent runs. Best values in bold; NN is deterministic (single run). RPD is computed from the best-known distance $D^* = 283.76\,\mathrm{km}$.

Algorithm	Best (km)	Mean (km)	Worst (km)	Std (km)	RPD (%)	CV (%)	Time (s)
MMAS	283.76	284.35	286.71	0.85	0.21	0.30	1.704
ACS	283.94	285.49	288.48	1.80	0.61	0.63	0.327
GA	291.31	302.13	318.03	6.79	6.47	2.25	0.676
SA	283.76	283.82	283.97	0.08	0.02	0.03	3.496
NN	297.10	297.10	297.10	0.00	4.70	0.00	0.001

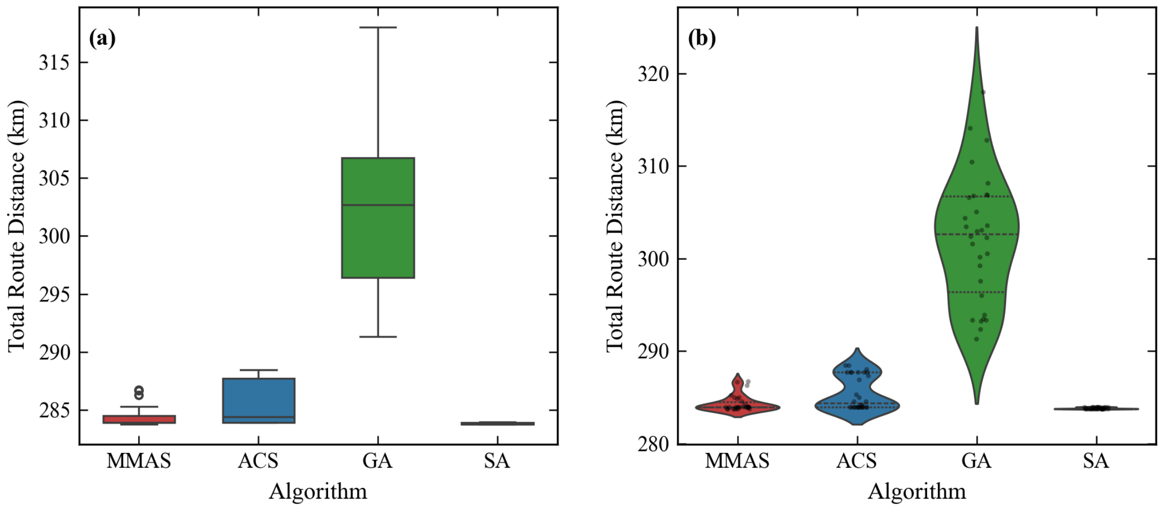


Fig. 2: Distribution of best distances across 30 runs. Box plots show median, interquartile range, and outliers; violin overlays show kernel density.

TABLE III: Wilcoxon signed-rank test results (significance level 0.05). All six pairwise comparisons are statistically significant.

Pair	W	p-value	Pair	W	p-value
MMAS vs ACS	68.0	0.011	ACS vs GA	0.0	< 0.0001
MMAS vs GA	0.0	< 0.0001	ACS vs SA	9.0	< 0.0001
MMAS vs SA	31.0	< 0.0001	GA vs SA	0.0	< 0.0001

B. Convergence Behaviour

Fig. 3 plots the mean best-so-far distance over 500 iterations, averaged across the 30 runs. The four algorithms display markedly different convergence profiles, as the figure shows.

The SA curve in Fig. 3 begins with the poorest initial solutions due to its random permutation start, but it descends the fastest and reaches its final value by roughly iteration 223, less than half the allocated budget. The 2-opt neighbourhood generates large improving moves early in the cooling schedule, while the Metropolis acceptance criterion (Eq. 4) preserves enough randomness to escape local optima before the temperature drops.

As Fig. 3 further illustrates, MMAS follows a steadier trajectory. The bounded pheromone trail mechanism prevents premature convergence, allowing the colony to continue improving until around iteration 357 before the trails fully consolidate. Compared to SA, this colony-based search was slower to refine the final solution, but the bounded trails ensured that no run stagnated early.

The ACS curve shows the fastest initial descent among the ACO variants, cutting over $15\,\mathrm{km}$ from its starting tour in the first 50 iterations. However, it plateaus around iteration 178, much sooner than MMAS, with little subsequent improvement. The high exploitation probability (0.9) drives aggressive early exploitation but leaves insufficient diversity for later refinement.

GA converged the most slowly, as the rightmost portion of Fig. 3 confirms. Its initial solutions were substantially longer than those of the other algorithms because it uses purely random permutations without a nearest-neighbour seed. More importantly, GA was still improving at the final generation (iteration 500), suggesting that the allocated budget was insufficient and that hybridisation with local search operators could narrow the quality gap.

C. Optimised Routes

Fig. 4 compares the best routes obtained by each algorithm, plotted over the road network of Yogyakarta. As Fig. 4a and 4b show, the optimal tour ($283.76\,\mathrm{km}$, found by both SA and MMAS) traces a geographically efficient loop that avoids

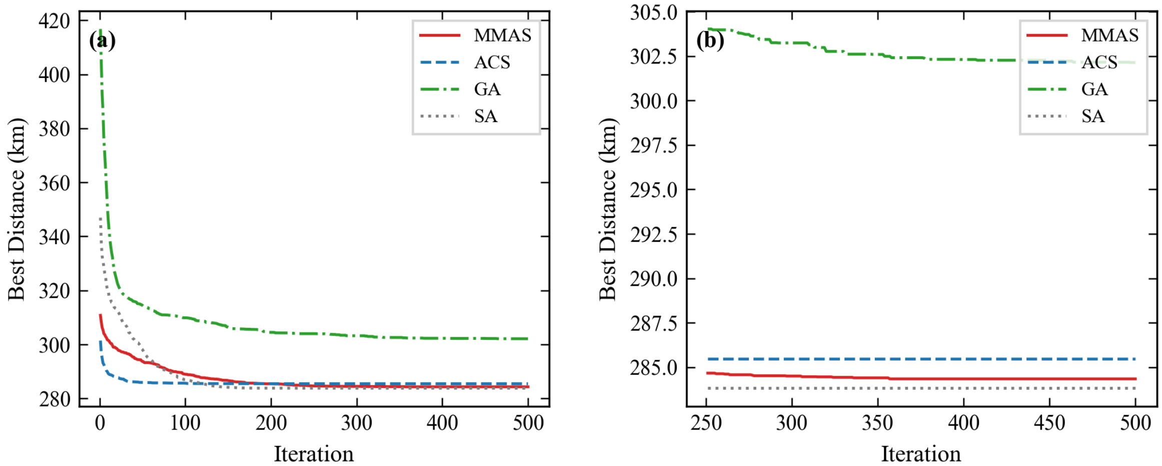


Fig. 3: Convergence curves showing mean best-so-far distance over iterations (averaged across 30 runs). SA converges fastest to the lowest value; GA is still improving at iteration 500.

crossing legs. Starting from the urban core (Kraton, Taman Sari, Alun-Alun Kidul), the route heads southeast through Purawisata and Gembira Loka Zoo to Kotagede, then swings south to the coastal zone (Parangtritis, Pantai Indrayanti, Goa Pindul). From the coast, it climbs northeast through the forested hills (Hutan Pinus Mangunan, HeHa Sky View) to the eastern temple complex (Ratu Boko, Tebing Breksi, Prambanan), continues north to Museum Ullen Sentalu in the highland, crosses west to Candi Borobudur, and returns through northern Yogyakarta (Monumen Jogja Kembali, Museum Affandi, Tugu, Malioboro) before closing the loop. This geographic clustering naturally minimises backtracking across the region's distinct tourism zones.

The ACS route in Fig. 4c followed a nearly identical sequence, differing by only one swap in the urban core segment and adding less than $0.2\,\mathrm{km}$ to the total distance. Fig. 4d reveals that GA's best tour contained visible detours where the route jumped between non-adjacent zones instead of completing one geographic cluster before moving to the next, a typical weakness of OX crossover, which preserves relative ordering but does not directly optimise edge connections.

D. Parameter Sensitivity

To assess how sensitive MMAS is to its control parameters, 27 configurations were evaluated (10 runs each) over a factorial grid of pheromone weight, heuristic weight, and evaporation rate. Fig. 5 presents the resulting heatmaps, where each cell corresponds to the mean distance for a given combination. The colour gradient in Fig. 5 indicates that the best configuration (pheromone weight 2.0, heuristic weight 2, evaporation rate 0.02) achieved a mean distance of $283.82\,\mathrm{km}$, matching SA's performance reported in Table II, while the worst configuration yielded $290.22\,\mathrm{km}$, a $6.4\,\mathrm{km}$ gap from parameter choice alone.

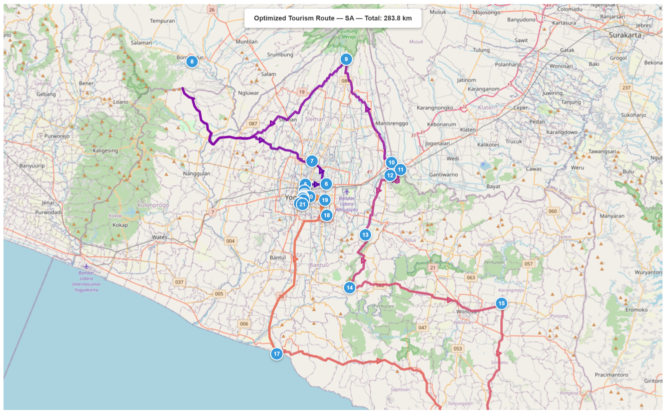
Comparing the three panels in Fig. 5 reveals that the interaction between pheromone weight and evaporation rate was the dominant factor. When pheromone weight is high (right panel), low evaporation works best because the colony can steadily reinforce high-quality edges over many iterations. When pheromone weight is low (left panel), the trails carry less information and need to be refreshed more frequently; higher evaporation rates compensate by clearing outdated pheromone and encouraging re-exploration. Simply raising the evaporation rate from 0.02 to 0.10 at the lowest pheromone weight reduced the mean distance by $2.3\,\mathrm{km}$, as the left panel of Fig. 5 shows.

By contrast, the heuristic weight had a comparatively minor influence. Moving vertically within any single panel of Fig. 5 (i.e., changing the heuristic weight from 2 to 5) rarely shifted the mean distance by more than $1\,\mathrm{km}$. One exception occurred at moderate pheromone weight with high heuristic weight and mid-range evaporation, where all 10 runs converged to the identical tour with zero variance, indicating that a strong distance heuristic can force complete convergence at the expense of solution diversity.

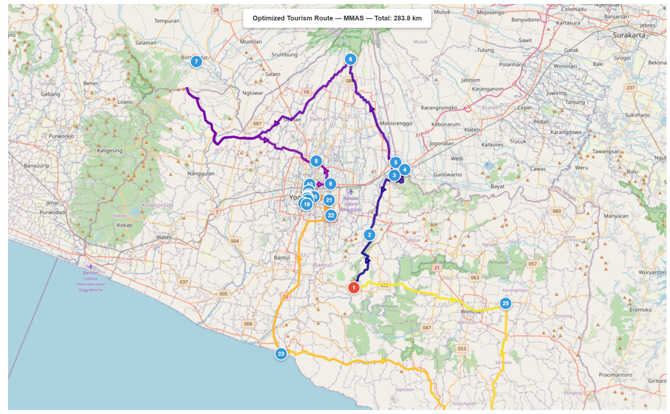
E. Scalability

Fig. 6 reports the mean distance and computation time for subsets of 10, 12, 15, and 20 attractions (10 runs each), alongside the full 25-attraction results from Table II. The top panel of Fig. 6 shows that at small sizes (12 or fewer attractions), all four metaheuristics found the optimal tour in every run. GA also reached the optimum in its best runs but already showed non-zero variance at 12 attractions, foreshadowing its later difficulties.

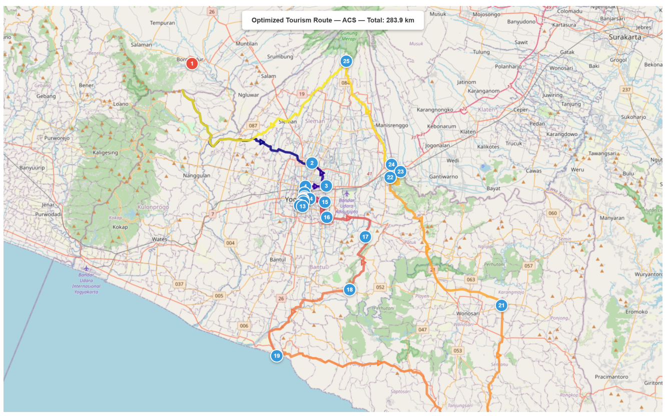
As the distance curves in Fig. 6 illustrate, the algorithms began to separate at 15 attractions. While MMAS, ACS, and SA continued to find the optimum reliably, GA's mean rose to 3.1% above the best-known solution, and the NN baseline diverged even further. At 20 attractions, SA still produced near-perfect results, MMAS and ACS remained within 0.2% of the optimum, but GA's gap grew to 5.2%. This widening continued at the full 25-attraction problem (see Table II), where GA trailed by 6.5%.



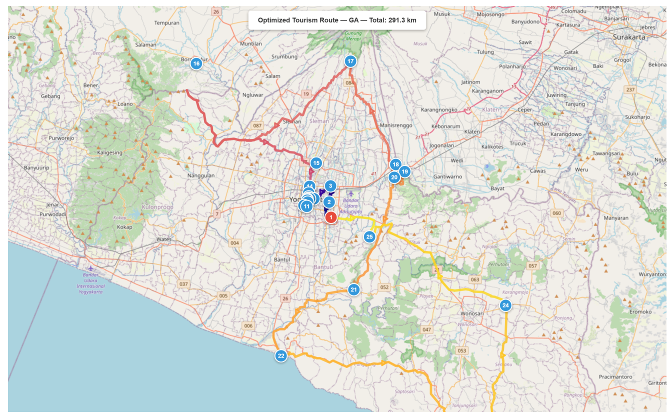
(a) SA — 283.76 km



(b) MMAS — 283.76 km



(c) ACS — 283.94 km



(d) GA — 291.31 km

Fig. 4: Best routes found by each algorithm on the road network. Green marker: start; red: end; blue: intermediate stops. SA and MMAS share the same optimal tour structure; GA shows visible detours between geographic clusters.

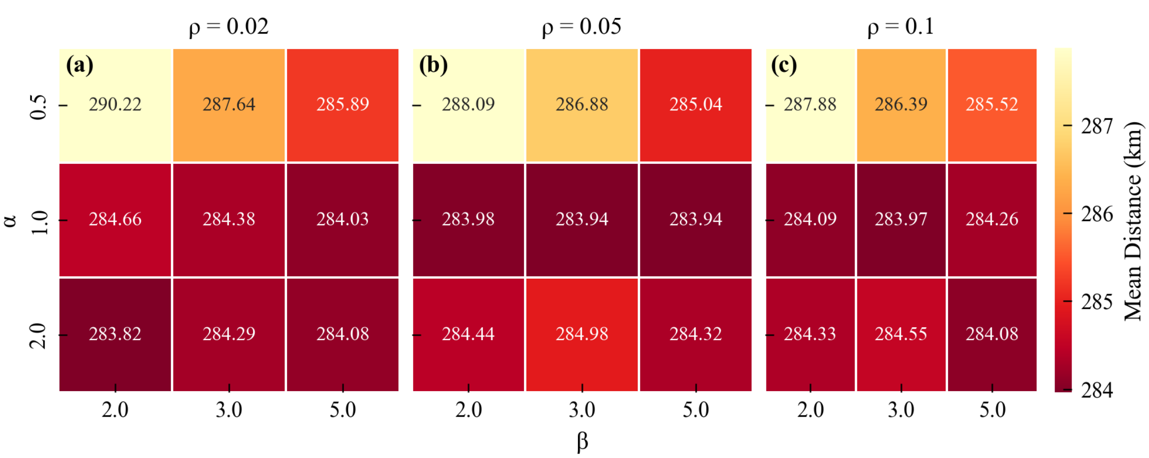


Fig. 5: MMAS parameter sensitivity heatmaps. Each cell shows the mean distance (km) over 10 runs for a given heuristic weight and evaporation rate combination at three levels of pheromone weight. The best configuration (pheromone weight 2.0, heuristic weight 2, evaporation rate 0.02) matches SA's mean of 283.82 km.

The bottom panel of Fig. 6 shows the corresponding runtime behaviour. MMAS and ACS exhibited approximately quadratic growth with problem size, consistent with their pheromone update mechanism. GA scaled roughly linearly. SA's runtime stayed nearly constant at about $3.5\,\mathrm{s}$ regardless of problem size, since its computation is governed by the cooling schedule rather than the number of cities.

F. Euclidean Versus Road Distance

A central motivation for this study is that Euclidean distances are an imperfect proxy for actual travel. To quantify this, we computed the detour ratio for each attraction pair:

$$r_{ij} = \frac{d_{\mathrm{road}}(i,j)}{d_{\mathrm{euclid}}(i,j)} \tag{7}$$

where a ratio of 1 would mean a perfectly straight road.

Fig. 7 visualises the comparison across all 300 attraction pairs. The scatter plot on the left side of Fig. 7 shows a strong linear correlation (R-squared of 0.97) between Euclidean and road distances, but with a consistent positive bias: road distances were on average 18% longer, with a mean ratio of 1.18 and a standard deviation of 0.12. The histogram on the right side of Fig. 7 shows the distribution of detour ratios. The ratio was lowest (around 1.05) for pairs connected by direct highway segments, such as the Yogyakarta–Prambanan corridor, and highest (up to 1.72) for pairs separated by rivers, mountain slopes, or dense urban fabric. Although the overall correlation is high, the consistent positive bias means that the errors compound across a multi-stop tour. For the 25-attraction problem in Table I, relying on Euclidean distances would underestimate the total tour length by roughly $40 - -50\,\mathrm{km}$, enough to render a planned day trip infeasible in practice.

G. Discussion

Algorithm selection trade-offs. The results in Table II and the convergence profiles in Fig. 3 together inform algorithm selection for different application contexts. SA with 2-opt is the strongest option when a few seconds of computation time is acceptable, as in offline trip planning tools or pre-computed itinerary brochures. ACS is better suited to interactive applications, such as a mobile app that recalculates routes in real time, because it runs an order of magnitude faster (Table II, Time column) while producing tours that are only marginally longer. MMAS is a middle option: more reliable than ACS (Table III, p = 0.011), but slower. GA with basic OX crossover is not competitive at this problem size, as Fig. 6 confirmed across multiple problem sizes, though hybridising it with local search operators [17] or edge-assembly crossover [6] would likely improve its performance.

Practical implications. The optimal $283.76\,\mathrm{km}$ tour reported in Table II translates to roughly 7 hours of driving at Yogyakarta's typical mixed-road speed of $40\,\mathrm{km/h}$. Combined with 30–60 minutes spent at each attraction, visiting all 25 sites listed in Table I would span 3–4 days. The geographic clustering visible in the optimal route (Fig. 4a) suggests a natural breakdown into daily itineraries: (1) the urban core (Kraton, Malioboro, Taman Sari, and surrounding sites), (2) the eastern temple complex (Prambanan, Ratu Boko, Tebing Breksi) combined with the northern highland (Ullen Sentalu), (3) the southern coast (Parangtritis, Indrayanti, Goa Pindul, Hutan Pinus), and (4) a western excursion to Borobudur. Tour operators could use these clusters as ready-made day-trip modules.

Comparison with related work. The 18% average road-to-Euclidean deviation shown in Fig. 7 is consistent with findings by Boyaci et al. [8] for European vehicle routing and by Tatit et al. [26] for spatial query accuracy, confirming that the bias is not unique to Yogyakarta. Unlike Sun et al. [23] and Nasution et al. [27], who used simplified distance metrics, the present study shows that real road distances can alter both absolute tour lengths and the relative ranking of near-optimal solutions. The parameter sensitivity results in Fig. 5 are also consistent with the general tuning guidance of Kaushik and Nadeem [30], who stressed that ACO parameters interact and should not be tuned in isolation.

Limitations. This study optimises total distance only. In practice, travellers also care about travel time, entrance fees, opening hours, and personal preferences; multi-objective formulations [20], [23] would capture these factors more faithfully. The road distances used here are static and do not reflect traffic congestion or time-of-day variations. The analysis is limited to a single destination with 25 attractions; testing on larger instances or other Indonesian cities would strengthen the generalisability of the findings. Finally, orienteering formulations with time windows [22], [28] and state-of-the-art exact or near-exact solvers such as LKH [9] were not included in the comparison and represent natural extensions.

V. Conclusion

This study compared four metaheuristic algorithms—MMAS, ACS, GA, and SA—for a 25-attraction tourism route optimisation problem in Yogyakarta, formulated as a TSP over real road distances extracted from OpenStreetMap. Three principal findings emerged. First, SA with 2-opt consistently produced the shortest and most reliable tours, achieving a mean of $283.82\,\mathrm{km}$ across 30 runs with a standard deviation below $0.1\,\mathrm{km}$. All pairwise differences were statistically significant ($p < 0.05$), yielding a clear ranking of SA, MMAS, ACS, and GA from best to worst. Second, ACS offered the best speed–quality trade-off, completing runs an order of magnitude faster than SA while producing tours only marginally longer, making it suitable for real-time applications. Third, the mean road-to-Euclidean detour ratio of 1.18 confirms that Euclidean distances systematically underestimate travel costs, reinforcing the need for real road network data in tourism route planning.

Parameter sensitivity analysis revealed that MMAS performance varies by up to $6.4\,\mathrm{km}$ depending on parameter settings, with a strong interaction between the pheromone weight and evaporation rate. Scalability tests showed that GA's quality gap widens with problem size, while the other three algorithms remain robust up to 25 attractions.

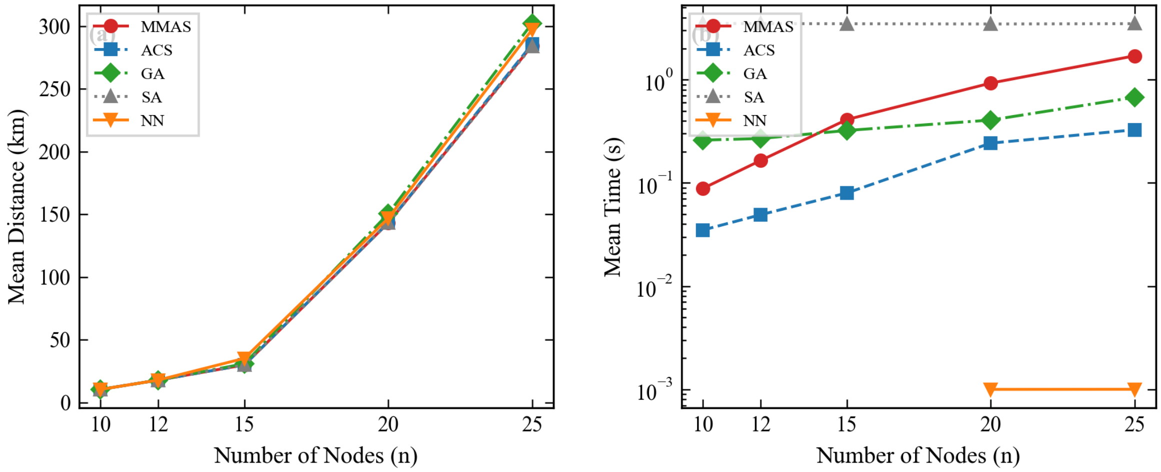


Fig. 6: Scalability analysis for 10, 12, 15, 20, and 25 attractions. Top: mean distance; bottom: mean computation time. GA diverges from the other algorithms as problem size increases; SA runtime remains constant.

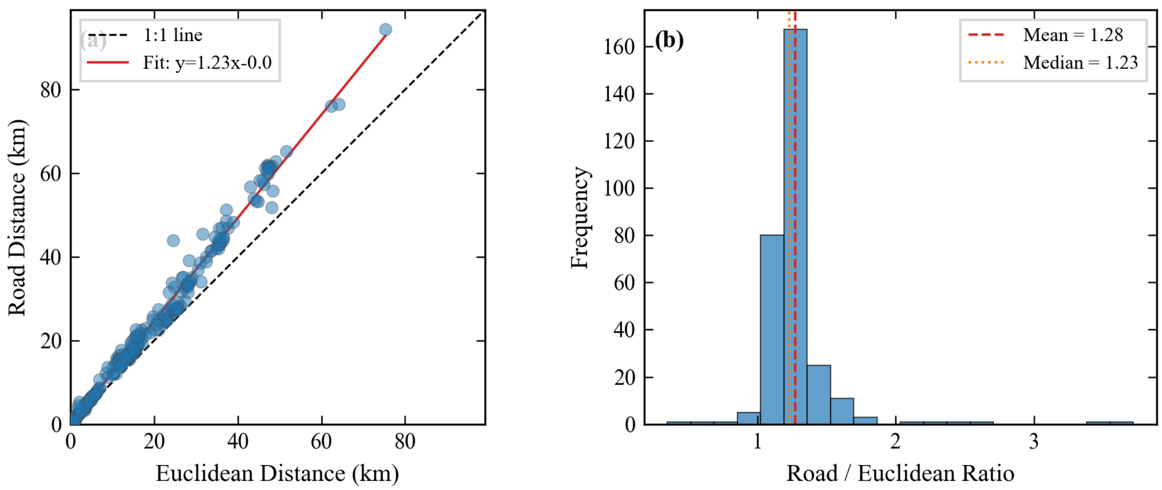


Fig. 7: Euclidean versus road distance comparison. Left: scatter plot with linear regression (R-squared = 0.97); right: histogram of detour ratios (mean 1.18, range 1.05–1.72).

Future work should extend this framework to multi-objective formulations that account for travel time, visitor preferences, and time-window constraints, and should incorporate dynamic traffic conditions. Validating the approach on other Indonesian tourism destinations would further establish its practical utility.

Declaration of Competing Interest

The authors declare no known competing financial interests or personal relationships that could have influenced this work.

Data Availability

The source code, distance matrices, and experimental results are available at [repository URL].

References

[1] Badan Pusat Statistik DIY, "Statistik kepariwisataan daerah istimewa yogyakarta," BPS Provinsi D.I. Yogyakarta, Tech. Rep., 2023.

[2] B. Toaza and D. Esztergar-Kiss, "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems," *Applied Soft Computing*, vol. 148, p. 110908, 2023.

[3] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13 187–13 257, 2023.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[5] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[6] Y. Deng, J. Xiong, and Q. Wang, "A hybrid cellular genetic algorithm for the traveling salesman problem," *Mathematical Problems in Engineering*, vol. 2021, p. 6697598, 2021.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] B. Boyaci, T. H. Dang, and A. N. Letchford, "Vehicle routing on road networks: how good is Euclidean approximation?" *Computers & Operations Research*, vol. 129, p. 105197, 2021.

[9] J. Zheng, K. He, J. Zhou, Y. Jin, and C.-M. Li, "Reinforced Lin-Kernighan-Helsgaun algorithms for the traveling salesman problems," *Knowledge-Based Systems*, vol. 260, p. 110144, 2023.

[10] M. Yousefikhoshbakht, “Solving the traveling salesman problem: a modified metaheuristic algorithm,” *Complexity*, vol. 2021, p. 6668345, 2021.

[11] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[12] P. Du, N. Liu, H. Zhang, and J. Lu, “An improved ant colony optimization based on an adaptive heuristic factor for the traveling salesman problem,” *Journal of Advanced Transportation*, vol. 2021, p. 6642009, 2021.

[13] R. Skinderowicz, “Improving ant colony optimization efficiency for solving large TSP instances,” *Applied Soft Computing*, vol. 120, p. 108653, 2022.

[14] T. Fei, X. Wu, L. Zhang, Y. Zhang, and L. Chen, “Research on improved ant colony optimization for traveling salesman problem,” *Mathematical Biosciences and Engineering*, vol. 19, no. 8, pp. 8152–8186, 2022.

[15] T. Hao, Y. Wu, J. Zhang, and J. Zhang, “Study on a hybrid algorithm combining enhanced ant colony optimization and double improved simulated annealing via clustering in the TSP,” *PeerJ Computer Science*, vol. 9, p. e1609, 2023.

[16] S. Cao, “An optimal round-trip route planning method for tourism based on improved genetic algorithm,” *Computational Intelligence and Neuroscience*, vol. 2022, p. 7665874, 2022.

[17] Y. Ruan, W. Cai, and J. Wang, “Combining reinforcement learning algorithm and genetic algorithm to solve the traveling salesman problem,” *Journal of Engineering*, vol. 2024, no. 6, p. e12393, 2024.

[18] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[19] P. H. Gunawan and Iryanto, “Simulated annealing – 2 opt algorithm for solving traveling salesman problem,” *International Journal of Computing*, vol. 22, no. 1, pp. 43–50, 2023.

[20] J. Ruiz-Meza and J. R. Montoya-Torres, “A systematic literature review for the tourist trip design problem: extensions, solution techniques and future research lines,” *Operations Research Perspectives*, vol. 9, p. 100228, 2022.

[21] J. Zhong, X. Wang, and L. Li, “Optimization for the multiday urban personalized trip design problem with time windows and transportation mode recommendations,” *Transportation Research Record*, vol. 2677, no. 5, pp. 327–340, 2023.

[22] K. Sylejmani, V. Abdurrahmani, A. Ahmeti, and E. Gashi, “Solving the tourist trip planning problem with attraction patterns using metaheuristic techniques,” *Information Technology & Tourism*, vol. 26, no. 4, pp. 633–678, 2024.

[23] H. Sun, Y. Chen, J. Ma, Y. Wang, X. Liu, and J. Wang, “Multi-objective optimal travel route recommendation for tourists by improved ant colony optimization algorithm,” *Journal of Advanced Transportation*, vol. 2022, p. 6386119, 2022.

[24] G. Boeing, “Modeling and analyzing urban networks and amenities with OSMnx,” *Geographical Analysis*, vol. 57, no. 4, pp. 567–577, 2025.

[25] ——, “Street network models and indicators for every urban area in the world,” *Geographical Analysis*, vol. 54, no. 3, pp. 519–535, 2022.

[26] P. Tatit, K. Adhinugraha, and D. Taniar, “Navigating the maps: Euclidean vs. road network distances in spatial queries,” *Algorithms*, vol. 17, no. 1, p. 29, 2024.

[27] S. M. Nasution *et al.*, “Tourism itinerary recommendation using vehicle routing problem time windows and analytics hierarchy process,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 36, no. 1, pp. 517–534, 2024.

[28] D. H. Fathurrohman, G. Alberto, and T. Iswari, “Application of green orienteering problem for tourism case in Yogyakarta,” *Systemic Analytics*, vol. 3, no. 4, pp. 270–285, 2025.

[29] S. Thipsingh *et al.*, “Social and sustainable determinants of the tourist satisfaction and temporal revisit intention: a case of Yogyakarta, Indonesia,” *Cogent Social Sciences*, vol. 8, no. 1, p. 2068269, 2022.

[30] D. Kaushik and M. Nadeem, “Parameter tuning in metaheuristics: a bibliometric and gap analysis,” *International Journal of Information Technology*, vol. 16, pp. 1645–1651, 2024.