JOURNAL OF COMPUTATIONAL INNOVATION AND ANALYTICS
https://e-journal.uum.edu.my/index.php/jcia

How to cite this article:
Chew, J. V. L. & Sainin, S. M. S. (2026). Ant Colony Optimization for Efficient Emergency Ambulance Routing in Urban Environments. *Journal of Computational Innovation and Analytics*, *5*(1), 41-50. https://doi.org/10.32890/jcia2026.5.1.3

Ant Colony Optimization for Efficient Emergency Ambulance Routing in Urban Environments

[1]Jackel Vui Lung Chew & [2]Samry Mohd Shamrie Sainin
[1,2]Faculty of Computing and Informatics,
Universiti Malaysia Sabah, Malaysia

[1]Corresponding author: jackelchew93@ums.edu.my

Received: 20/05/2025 Revised: 25/07/2025 Accepted: 22/01/2026 Published: 31/01/2026

ABSTRACT

Efficient ambulance routing plays an important role in emergency medical services. However, solving the ambulance routing problem remains challenging. This study investigates the performance of ant colony optimization to solve the ambulance routing problem, aiming to improve the quality of route planning under constraints such as traffic, patient urgency, and ambulance capacity. To simulate realistic emergency scenarios, 27 benchmark instances from the classical vehicle routing problem were adapted to the ambulance routing context by mapping depots to ambulance stations, customer nodes to emergency sites, and incorporating patient urgency and ambulance capacity. The performance of ant colony optimization was compared with the genetic algorithm and particle swarm optimization. Each algorithm was independently applied to all instances, and route quality was evaluated based on best route cost, average cost, and standard deviations. The experimental results show that ant colony optimization consistently outperformed both genetic algorithm and particle swarm optimization across most instances. Specifically, ant colony optimization achieved shorter total route distances, which are measured as the cumulative route distance of ambulances required to serve the emergency sites. These improvements were accompanied by greater consistency in solution quality across multiple runs. These findings suggest that ant colony optimization is a robust and effective tool for ambulance routing optimization. This study contributes to the growing body of work on intelligent emergency logistics by demonstrating the practical advantages of ant colony optimization in critical decision-making. The findings are valuable for optimization researchers in enhancing ambulance routing efficiency.

Keywords: ant colony optimization, emergency medical services, genetic algorithm, particle swarm optimization, vehicle routing problem.

INTRODUCTION

Efficient ambulance routing plays an important role in emergency medical services, where timely arrival at the scene can be the difference between life and death. Determining near-optimal routes in real time is crucial for reducing ambulance response times, improving patient survival rates, and optimizing the allocation of emergency resources. As urban environments become increasingly complex, the need for intelligent ambulance routing strategies has grown more urgent.

However, solving the ambulance routing problem (ARP) presents significant challenges. Urban traffic congestion often leads to delays that can adversely affect patient outcomes (Poulton et al., 2018; Nagamani & Bhuvaneswari, 2022). The necessity to account for dynamic traffic conditions, particularly during peak and off-peak hours, further complicates the ambulance routing process. Moreover, the varying urgency levels of patients demand adaptive routing strategies, making the optimization problem even more complex (Sutherland & Chakrabortty, 2023; Benmessaoud & Ezziyyani, 2024).

This study addresses these challenges by investigating the performance of metaheuristics for ambulance routing. By integrating real-world constraints such as traffic congestion, patient urgency, and ambulance capacity into benchmark vehicle routing problem (VRP) instances, our approach captures the nature of ambulance routing in urban environments. Depots are mapped to ambulance stations, customer nodes to emergency sites, and patient urgency levels are encoded to influence route prioritization. The metaheuristic algorithm is designed to simulate intelligent agent behavior that adapts to traffic conditions with different complexities. Additionally, by considering ambulance capacity constraints, the ARP model ensures efficient allocation of emergency resources.

Several metaheuristic algorithms, such as genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO), have demonstrated potential in tackling routing problems (Zhao et al., 2015; Wu, 2017; Zhang & Xiong, 2018; Mouhcine et al., 2018; Nozari et al., 2025). Nonetheless, each of these methods has inherent limitations: GA may exhibit slow convergence, PSO is susceptible to getting trapped in local optima, and ACO can suffer from premature convergence. These challenges highlight the importance of exploring more robust optimization approaches.

To evaluate metaheuristic performance in ambulance routing, this study applies GA, PSO, and ACO independently to adapted benchmark instances from the classical VRP dataset proposed by Augerat et al. (1995). While previous research has explored heuristic approaches for ARP (Tlili et al., 2017; Issam et al., 2019), there is no evidence of ACO being applied directly to the Augerat benchmark in the context of ambulance routing. To this end, our study contributes a novel adaptation of ACO to this context. Although we use a standard ACO framework, its integration with urban ARP constraints, specifically traffic, patient urgency, and ambulance capacity, provides a practical enhancement over conventional applications. The comparative analysis of GA, PSO, and ACO in this benchmark setting provides new insights into their capability for emergency response optimization.

RELATED WORKS

ARP is a specialized extension of VRP that plays a crucial role in the operational efficiency of emergency medical services. Unlike the classical VRP, the ARP involves life-critical decisions that demand rapid and adaptive routing solutions. Its complexity is heightened by factors such as traffic congestion, dynamic road conditions, and varying levels of patient urgency. According to Lakshmi et al. (2023), the ARP encompasses two main phases: the response phase, where an ambulance is dispatched to the emergency site, and the transport phase, where the patient is transferred to the nearest appropriate hospital. Effective optimization must address both phases to minimize the total emergency response time and ensure timely medical intervention.

To enable the application of metaheuristic optimization, classical VRP benchmark instances are frequently adapted to ARP scenarios. This is done by mapping depots to ambulance stations, customer nodes to emergency sites, demands to levels of patient urgency, vehicle capacity to ambulance capacity, and the distance matrix to inter-site distances (Mouhcine et al., 2018; Issam et al., 2019; Hanawy et al., 2023; Haddad et al., 2024).

Among the various metaheuristic methods, GA and PSO have been widely adopted to solve ARP-related problems. For example, Nozari et al. (2025) applied GA to optimize an ARP model incorporating spatial coordinates, response time, and transportation cost. Zhao et al. (2015) proposed an improved PSO algorithm to solve a time-dependent shortest path problem for emergency vehicles, while another study employed PSO to optimize ambulance allocation on expressways (Wu, 2017). In terms of ACO applications, Zhang and Xiong (2018) enhanced ACO with artificial immune systems to improve routing efficiency for grain emergency vehicles, achieving significant improvements in time and cost. Mouhcine et al. (2018) also developed a distributed ACO model for determining optimal ambulance paths.

Despite their demonstrated effectiveness, these algorithms each present a specific drawback. GA may suffer from slow convergence, PSO is prone to local optima entrapment, and ACO may experience premature convergence or high sensitivity to parameters. To address such issues, researchers have explored hybrid metaheuristic strategies. For instance, a hybrid model combining simulated annealing (SA) and tabu search (TS) was proposed for static ARP, using a cluster-first, route-second approach in which K-means clustering is followed by SA-TS optimization (Issam et al., 2019). Another notable hybrid, GASA, an integration of GA with SA, was introduced to solve the emergency location routing problem, a variant of ARP (Nahavandi et al., 2022). The most recent related study proposed a hybrid TS and TS-based hyper-heuristic for solving ARP with time windows (Tlili et al., 2024).

Although these hybrid strategies have improved performance, several research gaps remain. Notably, most existing studies focus solely on hybrid approaches (Issam et al., 2019; Nahavandi et al., 2022; Tlili et al., 2024), while limited attention has been given to comparative evaluations of individual metaheuristics on benchmark-adapted ARP scenarios. Moreover, standard VRP datasets adapted to the ARP context have not been fully utilized for consistent benchmarking. These gaps highlight the need for more systematic evaluations of optimization strategies for ambulance routing.

METHODOLOGY

This study aims to evaluate and compare the performance of three metaheuristic algorithms (GA, PSO, and ACO) on solving ARP that adapts VRP datasets. Each algorithm is independently implemented using a consistent fitness evaluation framework and applied to the same benchmark instance.

The ARP considered in this study involves determining optimal ambulance routes from an ambulance station to a set of emergency sites with known patient urgency levels, under the constraint of limited ambulance capacity. The objective is to minimize the total route distance while ensuring that each site is visited exactly once, the urgency level on any route does not exceed the ambulance capacity, and all routes start and end at the ambulance station.

Then, using the benchmark instance, a distance matrix is constructed based on the Euclidean distance between all sites. Ambulance routes are represented as permutations of site indices, and their feasibility is validated by splitting each permutation into capacity-compliant sub routes. The fitness function calculates the total route cost by summing up the distances of all sub routes.

Next is the implementation of metaheuristic algorithms. The GA implementation initializes a population of randomly shuffled site sequences. Then, this algorithm evolves this population over a defined number of generations using the standard genetic process, such as selection (top-performing individuals are selected based on fitness), crossover (order-based crossover is applied by combining segments from parent solutions), and mutation (random swap mutation occurs with a small probability to maintain diversity). The best-performing solution across generations is returned as the GA result.

On the other hand, PSO is adapted for permutation-based search by initializing a swarm of site sequences. Each particle represents a candidate solution and is evaluated based on the fitness function. The particles are iteratively updated by applying random swaps to explore new solutions and retaining the best-known personal and global solutions. The global best particle after all iterations is selected as the PSO result.

Meanwhile, ACO constructs ambulance routes based on pheromone trails and heuristic desirability. Each ant builds a complete route by probabilistically selecting the next customer based on pheromone intensity (reflecting historical quality) and heuristic information (favoring shorter paths). Pheromone updates are performed after all ants complete their routes, incorporating an evaporation rate and quality-based reinforcement. The best route found over all iterations is taken as the ACO result.

To further enhance route quality, each algorithm's output undergoes the same two-stage local search. Firstly, the 2-opt technique is used to remove edge-crossings by reversing route segments to reduce route distance. Then, the swap-based local search is used to pairwise swap site positions to improve the route cost. This uniform post-processing ensures fairness in the final comparison of algorithm performance. The complete workflow is summarized in the pseudocode shown in Table 1.

Table 1

Implementation and Comparison of GA, PSO, and ACO with Two-Stage Local Search

```
Input: distance matrix, urgency vector, ambulance capacity, hyperparameters for GA, PSO, ACO

1. Run GA, PSO and ACO independently → GA_Route, PSO_Route, ACO_Route
2. Refine GA_Route, PSO_Route, ACO_Route using 2-opt and swap-based local search
3. Compute GA_Cost, PSO_Cost, ACO_Cost using fitness function
4. Repeat for five times

Output:
- Best, Average and Standard Deviations of GA_Cost, PSO_Cost, ACO_Cost
```

EXPERIMENTAL SETUP

This study utilizes one of the benchmark datasets from the VRP literature, which is Vrp-Set-A from Augerat et al. (1995). It is worth mentioning that this dataset can be obtained online at http://vrp.atd-lab.inf.puc-rio.br/index.php/en/. Although it is originally designed for logistics and goods distribution, this dataset can be adapted to represent ARP by mapping depots to ambulance stations, customer nodes to emergency sites, demands to patient urgency levels, vehicle capacity to ambulance capacity, and distance matrix to distance between sites (Tlili et al., 2017; Issam et al., 2019). This mapping enables the use of standard VRP instances as proxies for evaluating metaheuristic solutions in ambulance routing scenarios.

Before execution, this dataset undergoes a preprocessing stage to map the ARP context onto the VRP framework. This includes reassigning nodes, adjusting vehicle capacities to match typical ambulance constraints, and validating the feasibility of service routes under emergency constraints. Then, parameter tuning was conducted empirically for each metaheuristic algorithm. For GA, we used a population size of 50, a mutation rate of 0.02, and ran for 100 iterations. For PSO, we used 50 particles and the same number of iterations. For ACO, we employed standard settings with $\alpha = 1$, $\beta = 2$, $Q = 100$, and an evaporation rate of 0.5. These algorithm parameters were chosen based on our preliminary experiments to ensure that the numerical experiment can be conducted as fairly as possible.

For ACO, the reason why we chose $\alpha = 1$ and $\beta = 2$ is to balance the relative influence of pheromone trails and heuristic information, respectively. The pheromone deposit factor $Q = 100$ was to ensure a sufficient update magnitude for reinforcing good paths, while the evaporation rate of 0.5 was chosen to maintain a balance between exploration and exploitation. These values were validated through preliminary experiments to ensure a fair experiment against the other two algorithms.

Evaluation of algorithmic performance is based on the minimum total route cost achieved across five independent runs (Best Cost), the mean total route cost over five independent runs (Average Cost), the variability in route cost across runs (Std Dev), and the percentage gap (GAP%) given by Equation 1.

$$\mathrm{GAP} = \frac{\text{Average Cost} - \text{Best Known Cost}}{\text{Best Known Cost}} \times 100\%. \tag{1}$$

All experiments were conducted in a consistent environment using Python 3.13 on a Windows 11 64-bit operating system. The hardware setup included an AMD Ryzen 7 5700U CPU with 8GB RAM, ensuring experimental fairness across all experimental runs.

RESULTS AND DISCUSSION

To evaluate the effectiveness of GA, PSO, and ACO in solving the ARP, 27 classical benchmark VRP instances from Augerat et al. (1995) were adapted and tested. The results of best-obtained costs and their percentage gaps relative to the best-known solutions are summarized in Table 2. Additionally, Table 3 presents the average solution cost and standard deviation across five independent runs to assess the stability and consistency of each algorithm. Besides, Figure 1 is provided for a visual comparison of the three algorithms using the top five instances where ACO achieved the lowest average cost.

Table 2

Results of Best Cost and GAP% in Solving Augerat et al. (1995) Problem

Instance	Best Known	GA		PSO		ACO	
		Best	GAP%	Best	GAP%	Best	GAP%
A-n32-k5	784.00	969.99	36.11	1592.09	112.71	842.19	8.05
A-n33-k5	661.00	865.57	34.62	1290.86	105.49	674.25	4.79
A-n33-k6	742.00	945.76	30.67	1344.17	87.94	769.89	4.92
A-n34-k5	778.00	892.20	29.99	1499.88	95.51	812.86	5.14
A-n36-k5	799.00	1022.64	39.37	1568.21	101.67	855.16	8.78
A-n37-k5	669.00	902.05	44.51	1474.83	133.72	710.80	9.93
A-n37-k6	949.00	1256.94	35.38	1623.29	84.08	981.40	5.24
A-n38-k5	730.00	1073.50	52.48	1624.91	128.94	765.87	6.96
A-n39-k5	822.00	1069.08	43.46	1702.47	112.02	904.87	12.14
A-n39-k6	831.00	1157.35	47.10	1838.22	124.63	884.66	7.58
A-n44-k6	937.00	1293.60	45.42	1983.65	115.03	975.83	5.53
A-n45-k6	944.00	1492.92	63.42	2272.39	145.31	1040.39	10.53
A-n45-k7	1146.00	1632.03	45.47	2094.06	90.77	1226.87	8.00
A-n46-k7	914.00	1347.88	58.27	2048.29	131.73	1024.71	13.18
A-n48-k7	1073.00	1663.55	59.07	2257.73	119.04	1152.52	9.39
A-n53-k7	1010.00	1617.67	79.13	2526.66	153.60	1102.45	10.01
A-n54-k7	1167.00	1950.01	70.60	2604.44	133.10	1299.80	12.03
A-n55-k9	1073.00	1829.26	75.35	2537.20	139.69	1150.67	7.78
A-n60-k9	1354.00	2128.10	69.45	3054.28	127.78	1466.16	10.66
A-n61-k9	1034.00	1879.96	84.13	2596.27	156.14	1153.70	12.50
A-n62-k8	1288.00	2217.96	89.36	3150.82	149.34	1379.24	8.60
A-n63-k9	1616.00	2616.54	67.86	3443.39	118.48	1790.03	12.16
A-n63-k10	1314.00	2196.19	77.60	2989.78	131.90	1414.62	9.44
A-n64-k9	1401.00	2291.12	69.20	3114.58	126.15	1526.36	10.44
A-n65-k9	1174.00	2248.99	97.64	3160.22	178.97	1262.93	9.50
A-n69-k9	1159.00	2243.69	105.54	3133.32	181.94	1270.57	12.35
A-n80-k10	1763.00	3258.90	95.63	4185.99	142.00	1942.32	11.31

Based on the results shown in Table 2, ACO consistently achieved lower GAP% values compared to GA and PSO across most instances. For example, on instance A-n32-k5, ACO achieved a best cost of 842.19 with a GAP% of only 8.05%, significantly outperforming GA (GAP% = 36.11%) and PSO (GAP% = 112.71%). Similar patterns were observed for most instances, where ACO remained within 5-10% of the best-known solutions. In contrast, PSO exhibited the highest GAP% in nearly all instances, indicating poorer convergence in route quality. GA performed moderately better than PSO but was generally inferior to ACO in most cases. These results demonstrate the superior path construction and exploitation capabilities of ACO, which are beneficial in combinatorial routing problems. GA, while capable of finding feasible routes, exhibited higher deviations from optimality. Meanwhile, PSO struggled with premature convergence and local optima.

Table 3

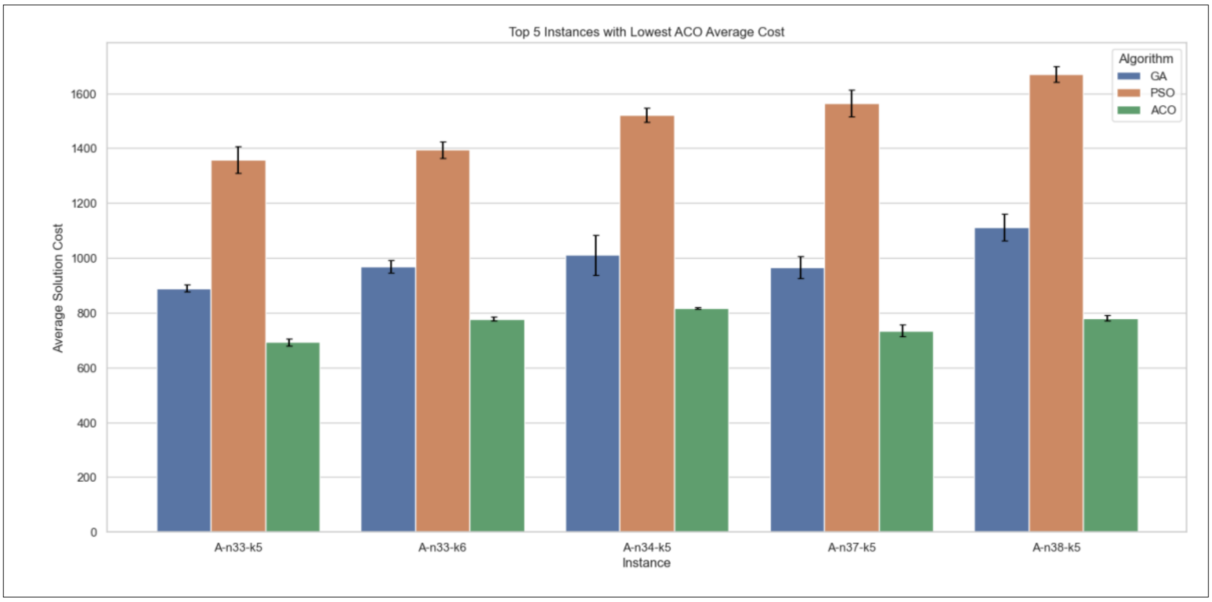
Average Cost and Std Dev in Solving Augerat et al. (1995) Problem

Instance	GA		PSO		ACO	
	Average	Std Dev	Average	Std Dev	Average	Std Dev
A-n32-k5	1067.09	57.91	1667.67	57.94	847.13	4.07
A-n33-k5	889.81	12.97	1358.28	49.75	692.68	12.87
A-n33-k6	969.58	23.03	1394.54	29.73	778.48	6.35
A-n34-k5	1011.33	72.52	1521.04	26.15	818.03	3.12
A-n36-k5	1113.58	69.15	1611.31	28.67	869.17	11.14
A-n37-k5	966.75	39.88	1563.62	48.38	735.43	21.53
A-n37-k6	1284.77	20.35	1746.95	75.80	998.72	9.27
A-n38-k5	1113.11	48.44	1671.27	29.14	780.84	10.46
A-n39-k5	1179.26	62.84	1742.83	30.78	921.77	11.22
A-n39-k6	1222.41	47.65	1866.66	24.09	894.02	7.39
A-n44-k6	1362.61	66.68	2014.82	22.46	988.79	9.08
A-n45-k6	1542.69	60.52	2315.72	45.87	1043.42	2.39
A-n45-k7	1667.05	33.75	2186.25	49.14	1237.71	8.85
A-n46-k7	1446.60	58.33	2117.97	43.65	1034.47	6.48
A-n48-k7	1706.78	24.08	2350.32	57.56	1173.78	12.00
A-n53-k7	1809.18	136.04	2561.40	39.41	1111.11	10.92
A-n54-k7	1990.95	28.66	2720.23	85.75	1307.36	4.04
A-n55-k9	1881.51	49.10	2571.91	32.03	1156.47	3.95
A-n60-k9	2294.34	102.94	3084.13	21.64	1498.29	16.72
A-n61-k9	1903.92	22.10	2648.48	36.69	1163.26	8.30
A-n62-k8	2438.99	127.74	3211.49	33.78	1398.77	12.06
A-n63-k9	2712.59	59.45	3530.56	55.74	1812.51	17.12
A-n63-k10	2333.70	79.48	3047.22	43.09	1438.10	14.65
A-n64-k9	2370.44	59.38	3168.41	31.72	1547.25	13.06
A-n65-k9	2320.24	43.72	3275.06	68.15	1285.55	15.74
A-n69-k9	2382.26	109.56	3267.70	91.25	1302.17	16.44
A-n80-k10	3448.87	104.82	4266.54	63.64	1962.47	18.94

Based on the results shown in Table 3, we found that ACO not only provided the best costs on average but also showed remarkable consistency, with low standard deviations across most instances. For instance, in A-n33-k5, ACO achieved an average cost of 692.68 with a standard deviation of only 12.87, compared to GA (Average = 889.81, Std Dev = 12.97) and PSO (Average = 1358.28, Std Dev = 49.75). Across larger instances such as A-n80-k10, ACO maintained a competitive average cost (1962.47) with the lowest variability (Std Dev = 18.94), whereas GA and PSO produced significantly higher costs (Average = 3448.87 and 4266.54, respectively) and wider variations. The results reinforce the robustness of ACO in producing consistent solutions.

Figure 1

Comparison of average solution costs and standard deviations of GA, PSO, and ACO on the top five benchmark instances with the lowest ACO performance



Based on Figure 1, we can observe the significance of performance comparison of GA, PSO, and ACO across the top 5 benchmark instances, with the lowest average cost achieved by ACO. For each instance, we found that ACO consistently outperforms both GA and PSO in terms of lower average solution cost, indicating superior optimization quality. Moreover, the error bars (standard deviations) for ACO are significantly smaller, suggesting that ACO not only finds better solutions but does so with higher stability and consistency across runs. In contrast, PSO not only yields the highest average cost but also exhibits larger variability, reflecting lower convergence reliability. GA performs moderately but is still notably less effective than ACO in most cases. This clustered bar graph strongly supports ACO as the most efficient and robust algorithm for adapted ARP instances.

The experimental results presented in this section were recorded from our fair experiment and comparison, where all three algorithms (GA, PSO, and ACO) were executed under a uniform computational budget with 50 individuals/particles/ants and a fixed number of 100 iterations. As mentioned in the previous section, the parameter settings were solely based on our preliminary experiments. While further tuning may enhance the performance of GA and PSO, parameters were deliberately fixed to isolate the intrinsic capabilities of each algorithm. This strategy ensures that observed differences stem from algorithmic behavior rather than from hyperparameter optimization, thereby reinforcing the validity of the comparative analysis.

Despite ACO's superior performance, we found several limitations in this algorithm. One of them is its relatively slower convergence speed in the early iterations for larger instances like A-n80-k10. This is partly due to its dependence on accumulated pheromone trails, which require several iterations to effectively guide the search process. Moreover, ACO's memory-dependent behavior may lead to search stagnation if exploration is not sufficiently maintained. Without adaptive mechanisms such as dynamic pheromone adjustment or hybridization with local search strategies, ACO may occasionally converge to suboptimal routes. These limitations suggest opportunities for future enhancement, such as integrating adaptive pheromone updating strategies.

CONCLUSION

This study evaluated and compared the performance of GA, PSO, and ACO in solving ARP by adapting 27 VRP instances from Augerat et al. (1995). The experimental results clearly indicate that ACO consistently outperforms GA and PSO in terms of both solution quality (lower cost and GAP%) and solution stability and consistency (lower standard deviation across multiple runs). GA showed moderate competitiveness but lacked consistency, while PSO generally performed the weakest, exhibiting high-cost solutions and significant variability. Among the algorithms tested, ACO demonstrated its ability to construct high-quality solutions through a distributed and cooperative search mechanism and its robustness in producing stable results across varied problem instances. However, ACO also showed limitations, particularly its slower initial convergence and potential for search stagnation in larger or more complex instances. These challenges can be addressed in future work through hybridization with adaptive pheromone updating strategies.

From a practical standpoint, EMS providers looking to integrate ACO into their ambulance dispatch systems can benefit from its flexibility in adapting to dynamic urban environments. To implement ACO effectively, providers should consider the following steps: (1) define precise operational constraints such as vehicle capacity, urgency levels, and station locations; (2) construct an accurate distance matrix from real-world traffic and road network data; (3) configure ACO parameters based on typical urban routing scenarios; and (4) integrate ACO within a decision support system that allows real-time updates and route adjustments. With these considerations, ACO can serve as a powerful decision-making tool to enhance responsiveness and resource utilization in EMS.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

Augerat, P., Naddef, D., Belenguer, J. M., Benavent, E., Corberan, A., & Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M, Université Joseph Fourier, Grenoble, France

Benmessaoud, Y., & Ezziyyani, M. (2024). Enhancing ambulance traffic movement: a data-driven approach. In M. Ezziyyani, J. Kacprzyk, & V. E. Balas (Eds.), Lecture Notes in Networks and Systems (vol 905, pp 407-412). Springer.

Haddad, A., Tlili, T., Nouaouri, I., & Krichen, S. (2024). Solving the multi-objective ambulance routing problem using NSGA III. 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), 2296-2300.

Hanawy Hussein, T. D., Frikha, M., & Rahebi, J. (2023). Harris hawks optimization for ambulance vehicle routing in smart cities. Eastern-European Journal of Enterprise Technologies, 122(2/3), 74-81.

Issam, Z., Al-Omani, M., & Aldhafeeri, K. (2019). A new approach based on the hybridization of simulated annealing algorithm and tabu search to solve the static ambulance routing problem. Procedia Computer Science, 159, 1216-1228.

Lakshmi, A. V., Sekhar, P. C., Joseph, K. S., & Priya, A. S. (2023). A systematic review of route optimization for ambulance routing problem. 2022 3rd International Conference on Bioinformatics and Data Science (ICBDS), 294-304.

Mouhcine, E., Karouani, Y., Mansouri, K., & Mohamed, Y. (2018). Toward a distributed strategy for emergency ambulance routing problem. 2018 4th International Conference on Optimization and Applications (ICOA), 1-4.

Nagamani, S., & Bhuvaneswari, V. (2022). An efficient multilevel framework for prediction of optimized ambulance routes using random forest classifier. International Journal of Intelligent Systems and Applications in Engineering, 10(2s), 148-156.

Nahavandi, B., Homayounfar, M., Daneshvar, A., & Shokouhifar, M. (2022). Hierarchical structure modelling in uncertain emergency location-routing problem using combined genetic algorithm and simulated annealing. International Journal of Computer Applications in Technology, 68(2), 150-163.

Nozari, H., Szmelter-Jarosz, A., & Irani, H. R. (2025). Designing an ambulance routing optimization model using the combination of machine learning and genetic algorithm in conditions of uncertainty. Systems and Soft Computing, 7, 200276.

Poulton, M., Noulas, A., Weston, D., & Roussos, G. (2018). Modeling metropolitan-area ambulance mobility under blue light conditions. IEEE Access, 7, 1390-1403.

Sutherland, M., & Chakrabortty, R. K. (2023). An optimal ambulance routing model using simulation based on patient medical severity. Healthcare analytics, 4, 100256.

Tlili, T., Harzi, M., & Krichen, S. (2017). Swarm-based approach for solving the ambulance routing problem. Procedia Computer Science, 112, 350-357.

Tlili, T., Ben Nasser, S., Chicano, F., & Krichen, S. (2024). Tabu search-based hyper-heuristic for solving the heterogeneous ambulance routing problem with time windows. International Journal of Intelligent Transportation Systems Research, 22(2), 446-461.

Wu, L. (2017). Research on expressway emergency vehicle allocation based on improved particle swarm optimization. In J. S. Pan, J. W. Lin, C. H. Wang, &X. Jiang, (Eds.), Advances in Intelligent Systems and Computing (vol. 536, 139-145). Springer.

Zhao, J. D., Duan, X. H., & Song, S. X. (2015). A novel particle swarm optimization algorithm for solving the shortest path problem in highway network. Advances in Transportation Studies, 2, 97-106.

Zhang, Q., & Xiong, S. (2018). Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. Applied Soft Computing, 71, 917-925.