International Journal on Robotics, Automation and Sciences

# Data-Driven Route Optimization for Large-Scale Transportation Systems

Ching Joe Tan and Khai Wah Khaw*

***Abstract* – The study investigates the use of data-driven optimization strategies to increase routing efficiency in large-scale transportation systems with realistic constraints on operation. The objectives are to combine network cost data with advanced optimization techniques, assess routing performance under capacity, time, and cost constraints. It also compares traditional shortest-path methods to heuristic and metaheuristic approaches. A computational framework was created to compare several algorithms on a large-scale transportation dataset from Kaggle while accounting for real-world constraints such as vehicle capacity and travel costs. Performance was evaluated using total routing cost and route efficiency. The findings demonstrate that metaheuristic techniques consistently outperform traditional algorithms in complicated, constraint-rich situations. Genetic Algorithms, in particular, achieved greater cost reduction and continued to perform well as the problem scale rose. Classical algorithms performed well only in simple circumstances. These findings emphasize the need of cost-effective, data-driven metaheuristic optimization in current logistics planning and encourage future study into hybrid and dynamic routing systems.**

***Keywords—Data-Driven Optimization, Route Optimization, Genetic Algorithms, Vehicle Routing Problem, Transportation System***

## I. INTRODUCTION

Transportation systems are essential for economic development, urban expansion, and worldwide supply chains. The fast expansion of e-commerce, urbanization, and cross-border trade has significantly raised the complexity of current transportation networks, emphasizing routing efficiency and reliability. Inefficient routing results in higher operational expenses, more fuel usage, longer delivery times, and negative environmental consequences [1]. As transportation networks expand to encompass a large number of nodes, vehicles, and operational limitations, classic routing approaches like manual scheduling or basic optimization become increasingly inefficient. As a result, route optimization has arisen as an important topic of research in transportation engineering, logistics management, and operations research. Advances in data analytics and computer power have allowed transportation networks to be represented as weighted graphs, with routing decisions based on measurable cost variables like distance, time, and fuel consumption. This data-driven methodology provides for more objective, scalable, and efficient route design than static or heuristic-based solutions [2].

Classical shortest-path algorithms, including Dijkstra's Algorithm and the A* algorithm, are widely used to determine optimal paths between nodes in a network. These algorithms provide mathematically optimal solutions under well-defined parameters, and they serve as a foundation for many navigation and routing applications. However, the use of them is primarily limited to point-to-point routing, making it difficult to apply to real-world transportation challenges involving multiple destinations, multiple vehicles, and complicated operating limitations. Large-scale transportation systems frequently involve fleet

*Corresponding Author email: khaiwah@usm.my, ORCID: 0000-0003-2646-6477
Ching Joe Tan is with the School of Management, Universiti Sains Malaysia, 11800 Penang, Malaysia. (email: chingjoe@student.usm.my)
Khai Wah Khaw is with the School of Management, Universiti Sains Malaysia, 11800 Penang, Malaysia. (email: khaiwah@usm.my)

International Journal on Robotics, Automation and Sciences (2026) 8, 2:99-105
https://doi.org/10.33093/ijoras.2026.8.2.12
Manuscript received: 16 Feb 2026 | Revised: 27 Mac 2026 | Accepted: 02 May 2026 | Published: : 31 Jul 2026

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

coordination, capacity management, and cost optimization over multiple stops, rendering accurate shortest-path solutions ineffective. To overcome these constraints, heuristic and metaheuristic algorithms have been developed to address complex routing issues, particularly variations of the Vehicle Routing Problem. The Clarke-Wright Savings Algorithm, Greedy Algorithms, Genetic Algorithms, and Ant Colony Optimization all prioritize scalability and adaptability, resulting in high-quality solutions for massive, constraint-rich systems in a reasonable amount of computational time.

The drive for this study originates from the requirement to evaluate routing algorithms within a unified, data-driven framework that takes into account realistic transportation constraints. Routing decisions must take into account various delivery points, vehicle limits, operating expenses, and computing efficiency. As network size increases, the solution space rapidly expands, making brute-force or accurate optimization methods impossible [3]. Although there has been substantial study on both classical routing algorithms and metaheuristic strategies, few studies have directly compared these methods within a single analytical framework. The importance of classical shortest-path methods as fundamental inputs for higher-level VRP optimization is frequently underestimated. This study fills this gap by combining shortest-path computation with heuristic and metaheuristic methods and carefully evaluating their combined performance on real-world transportation data.

The objectives of this study are to investigate how data analytics can improve route efficiency in large-scale transportation systems, to determine which routing algorithms perform best under real-world constraints, and to compare traditional optimization approaches to data-driven and metaheuristic methods. The purpose of this study is to identify optimal optimization algorithms for complicated routing issues by assessing performance, scalability, adaptability, and computing feasibility. Studying is significant since it contributes to both theory and practice. Academically, it presents a structured comparison of several algorithmic paradigms within a staged optimization framework, illustrating how classical shortest-path methods complement metaheuristic VRP techniques. In real life, the findings provide useful insights for transportation planners and logistics managers, showing the efficacy of adaptive, data-driven metaheuristic approaches particularly Genetic Algorithm-based methods in boosting routing efficiency and lowering operational costs.

This study distinguishes itself apart from previous VRP research by combining higher-level heuristic/metaheuristic vehicle routing techniques with traditional shortest-path algorithms into a unified data-driven framework. This study models baseline network-level travel efficiency using Dijkstra's and A* algorithms before using Clarke-Wright Savings, Greedy Multi-Vehicle, Genetic Algorithm, and Ant Colony Optimization techniques to solve fleet-level routing problems, in contrast to many previous studies that either concentrate on shortest-path computation or on vehicle routing heuristics alone. The study's layered structure makes feasible to examine how metaheuristic techniques enhance route consolidation, vehicle utilization, and operational cost under capacity, time, and cost limitations with how classical algorithms function as fundamental routing tools. Therefore, this study's contribution is not a novel algorithm but rather a comparative, data-driven framework that connects realistic VRP optimization with shortest-path modelling using an actual, open-source transportation dataset.

Despite its advantages, this study has several shortcomings. While sufficient for comparative study, the dataset employed is modest in comparison to national or global transportation networks, and it may not reflect all of the complexity of the real world. Furthermore, the analysis assumes static network costs, but real-world transportation networks are dynamic and influenced by traffic conditions, weather, and time-varying demand. As a result, the findings are more useful for strategic planning than real-time routing. Furthermore, metaheuristic algorithms are prone to parameter selection and stochastic processes, which can impair reproducibility across datasets. Future study should overcome these constraints by using larger datasets, dynamic network circumstances, and adaptive parameter tweaking to improve real-world applicability.

## II. METHODOLOGY

This study employs a quantitative, experimental research approach to evaluate and compare route optimization algorithms in a data-driven context. The methodology is computational in nature, emphasizing algorithm implementation, simulation, and performance benchmarking with structured transportation datasets. A comparison technique is used to determine how traditional shortest-path algorithms differ from heuristic and metaheuristic optimization approaches when applied to identical network conditions. The overall framework has a layered optimization design. Initially, shortest-path algorithms are employed to calculate baseline network efficiency and minimal travel distances. These results serve as reference benchmarks and inputs for more advanced routing methods. Later, heuristic and metaheuristic algorithms are used to handle more complex routing issues with many destinations and vehicles. All algorithms are written in Python to ensure consistency in data handling, model assumptions, and performance evaluation between experiments.

This study utilizes the publicly available Large-Scale Route Optimization dataset from Kaggle containing order, distance, and vehicle information. Order datasets specify delivery demand, including origin-destination pairs, weight, and area requirements, and are used to build a directed transportation network. Each location is represented as a node, and observed delivery links build directed edges, guaranteeing that the network accurately represents practical routing patterns. To increase numerical stability, weight and area values in bigger datasets are standardized but relative magnitudes remain unchanged. Distance data provide pairwise travel distances and are utilized as edge weights in routing computations. The dataset was used to construct the transportation network, where each unique source or destination was represented as a node and each distance record was represented as a weighted edge. After preprocessing and combining all unique source and destination locations, the final network consisted of 3,783 nodes, representing delivery or routing points. A separate truck dataset specifies vehicle capacity restrictions, dimensions, cost per kilometres, and travel speed, allowing for accurate fleet modelling. Location IDs are standardized during preprocessing, and datasets are combined into a uniform modelling framework to enable fair and consistent method evaluation.

In a few of aspects, the dataset is representative of actual transportation networks. First, it has a lot of linked origin-destination pairs, which makes it possible to build a sizable routing graph. Second, modelling trip costs and route efficiency across various delivery places is made possible by the use of pairwise distance data. Third, the dataset enables the analysis to take into account operational restrictions including fleet assignment, delivery demand, vehicle capacity, and route cost when combined with order and vehicle information. The dataset does have some restrictions, real-time traffic situations, weather effects, road closures, and stochastic service times are not directly included in this representation of a static transportation network. As a result, the dataset can be used to assess comparative algorithm performance and strategic route planning, but it is unable to depict the dynamic real-time transportation networks.

The routing problem in this study can be formulated as a capacitated vehicle routing problem with time and cost considerations [10]. The transportation network is represented as a directed weighted graph $G = (N, E)$, where $N$is the set of delivery nodes and $E$is the set of directed travel links between source and destination locations. Each edge $(i, j) \in E$has an associated travel distance $d_{ij}$and travel cost $c_{ij}$. Each customer node $i$has a delivery demand $q_i$, while each vehicle $k$has a maximum capacity $Q_k$.

The main decision variable is:

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels directly from node } i \text{ to node } j \\ 0, & \text{otherwise} \end{cases}$$

The objective of the optimization model is to minimize total routing cost:

$$\min Z = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N, j \neq i} c_{ij} x_{ijk}$$

where $K$represents the set of vehicles used. The model is subject to the following main constraints:

$$\sum_{k \in K} \sum_{i \in N, i \neq j} x_{ijk} = 1, \forall j \in N \setminus \{0\}$$

This ensures that each delivery node is visited once.

$$\sum_{i \in N} q_i \, y_{ik} \leq Q_k, \forall k \in K$$

This ensures that the total demand assigned to each vehicle does not exceed vehicle capacity.

$$\sum_{j \in N, j \neq i} x_{ijk} = \sum_{j \in N, j \neq i} x_{jik}, \forall i \in N, \forall k \in K$$

This flow conservation constraint ensures that if a vehicle enters a node, it must also leave that node.

$$a_i \leq t_i \leq b_i, \forall i \in N$$

This represents the time-window constraint, where $a_i$and $b_i$are the earliest and latest allowable service times at node $i$, and $t_i$is the vehicle arrival time.

In this study, exact mathematical optimization was not applied to solve the entire model since the huge solution space renders comprehensive optimization computationally impracticable [13]. Rather, the formulation offers the theoretical foundation for assessing heuristic and metaheuristic algorithms. The implemented algorithms seek to approximate high-quality feasible solutions that satisfy vehicle capacity, delivery feasibility, and time-related requirements while minimizing routing costs.

The optimization framework's baseline layer is based on shortest-path modeling. The transportation network is represented as a directed, weighted graph, with distances acting as edge weights. Two classical algorithms: Dijkstra's Algorithm and the A* algorithm are used to compute minimum distance pathways between sources and destinations. Duplicate edges are consolidated by keeping the shortest observed distance to reduce redundancy. Dijkstra's Algorithm is implemented using a priority queue and early termination to improve performance while computing the source to destination pathways [4]. Path reconstruction is achieved by precursor tracking, which allows whole route sequences to be retrieved. The A* algorithm extends this approach by including an acceptable heuristic obtained from reverse graph shortest path calculations [5]. This heuristic directs the search toward target nodes while maintaining optimality and eliminating wasteful exploration, hence increasing computational efficiency.

Heuristic and metaheuristic algorithms are used to solve multi-stop, multi-vehicle routing issues that exceed the capability of traditional shortest path approaches. The Clarke-Wright Savings Algorithm serves as a constructive heuristic. It initially assigning different routes to each destination and then repeatedly merging them depending on distance savings [6]. Time window feasibility is included to guarantee that delivery schedules are valid. The Greedy method is implemented as a fast-routing heuristic that incrementally chooses the next best possible destination based on distance, capacity, and time constraints. Neighbourhood trimming is used to improve scalability. While greedy decisions are locally optimal, this strategy achieves a good runtime to performance ratio and serves as a foundation for more advanced optimization strategies [7].

Metaheuristic optimization is carried out using Genetic Algorithm and Ant Colony Optimization techniques. The Genetic Algorithm employs permutation-based encoding and split-based decoding to construct feasible multi-vehicle routes while adhering to capacity and time constraints. Evolutionary operators such as selection, crossover, mutation, and elitism are used to explore the solution space and avoid local optima [8]. The Ant Colony Optimization algorithm generates solutions probabilistically using pheromone intensity and heuristic desirability, with pheromone updates guiding convergence across iterations. Following route construction, vehicle assignment occurs by selecting cost-effective truck types based on route demand and operational costs [9]. The final evaluation parameters include total distance, operational cost, fleet size, delivery feasibility, and vehicle utilization, allowing for a thorough algorithm comparison.

## III. Results

This chapter discusses the findings of implementing the six routing algorithms: Dijkstra, A*, Clarke-Wright Savings, Greedy Multi-Vehicle, Genetic Algorithm, and Ant Colony Optimization in a single experimental framework. Using the same transportation network, order dataset, preprocessing pipeline, constraint logic, and vehicle assignment rules allowed for a fair comparison of techniques. Total travel distance, operational cost, fleet size, route efficiency, on-time delivery rate, and computational behavior were all used to evaluate performance. Classical shortest-path techniques served as baselines for measuring network trip efficiency at the order level, while heuristic and metaheuristic VRP methods were tested for their capacity to combine deliveries and optimize fleet-level outcomes. The chapter concludes with a comparative analysis that emphasizes trade-offs between runtime and solution quality, proving Genetic Algorithms as the most effective approach for large-scale, constraint-driven route optimization in this study.

The delivery network structure in Figure 1 demonstrates a highly centralized topology. City_61 is the major hub, with direct links to most other cities, resulting in a star-shaped hub-and-spoke system. Peripheral nodes have few links, indicating that non-hub destinations can only be reached via direct transit. This structure implies that many orders originate at or are routed through City_61, which simplifies coordination while increasing reliance on a single crucial node.

Operationally, such concentration can lead to shortcomings such as congestion, delays, or interruptions impacting the hub may spread throughout the network. The presence of a few connecting nodes indicates that shortest-path routes will repeatedly pass through common intermediate towns. These network features warrant using shortest-path analysis to determine baseline connectivity and journey efficiency, as well as using VRP heuristics and metaheuristics to decrease redundant trips, improve consolidation, and increase resilience in a centralized system.

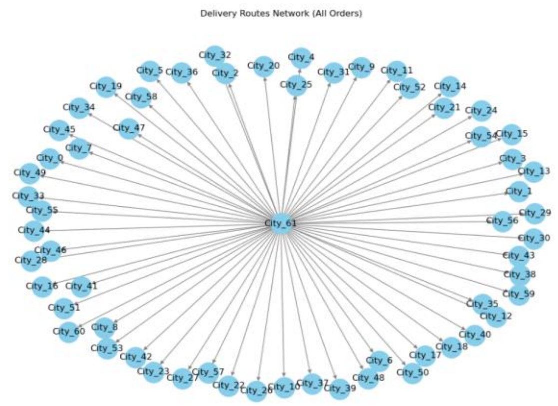


**FIGURE 1. Delivery network structure.**

Dijkstra's Algorithm was used to compute shortest-distance routes for each origin-destination pair, and it demonstrated deterministic and stable performance. Repeated orders with identical endpoints yielded the same shortest path, hop count, and travel distance, demonstrating consistent distance optimality based solely on edge weights. Figure 2. demonstrate the findings using Dijkstra's Algorithms. For example, deliveries from City_61 to City_54 consistently took the same multi-hop route and gave equal distance outputs, demonstrating that the algorithm's solution is unaffected by order volume or fleet context. The findings also emphasize the significance of intermediate hubs like City_49 and City_27, which appear often in reconstructed routes and are likely to serve as crucial links between the primary hub and surrounding cities. Despite providing order-level optimization, Dijkstra does not account for route consolidation, time windows, or shared vehicle capacity. Treating each order individually might result in overlapping trips, ineffective fleet usage, and higher operational costs on a scale.

| | Order_ID | Source | Destination | ShortestDistance(M) | Hops | Path |
|---|---|---|---|---|---|---|
| 0 | A140109 | City_61 | City_54 | 2106108 | 4 | City_61 → City_49 → City_27 → City_9 → City_54 |
| 1 | A140112 | City_61 | City_54 | 2106108 | 4 | City_61 → City_49 → City_27 → City_9 → City_54 |
| 2 | A140112 | City_61 | City_54 | 2106108 | 4 | City_61 → City_49 → City_27 → City_9 → City_54 |
| 3 | A140112 | City_61 | City_54 | 2106108 | 4 | City_61 → City_49 → City_27 → City_9 → City_54 |
| 4 | A140112 | City_61 | City_54 | 2106108 | 4 | City_61 → City_49 → City_27 → City_9 → City_54 |
| ... | ... | ... | ... | ... | ... | ... |
| 4640 | A240337 | City_61 | City_9 | 1554652 | 3 | City_61 → City_49 → City_27 → City_9 |
| 4641 | A240346 | City_61 | City_9 | 1554652 | 3 | City_61 → City_49 → City_27 → City_9 |
| 4642 | A270227 | City_61 | City_22 | 776681 | 2 | City_61 → City_49 → City_22 |
| 4643 | A270227 | City_61 | City_22 | 776681 | 2 | City_61 → City_49 → City_22 |
| 4644 | A160131 | City_61 | City_44 | 1207054 | 1 | City_61 → City_44 |

4645 rows × 6 columns

**FIGURE 2. Findings using dijkstra's algorithms.**

The A* Algorithm presented the same shortest distances, hop counts, and path sequences as Dijkstra for all corresponding origin-destination pairings, demonstrating that the heuristic was valid and did not overstate remaining cost. The reverse-graph heuristic directed the search to target nodes and avoided extraneous node expansions, increasing computational efficiency while maintaining optimality in large networks where exhaustive exploration might increase runtime. Similar to Dijkstra's findings, A* pathways repeatedly passed through crucial intermediate nodes, emphasizing the hub-and-spoke structure and the impact of densely connected transit cities on network traffic patterns. Yet, A* has the same structural limitation, it optimizes routes at the individual order level rather than the fleet level. It does not include vehicle capacity limits, delivery time windows, route consolidation, or fleet-wide cost minimization. As a result, A* is not appropriate as a stand-alone logistics routing solution, but it is still useful for generating precise distance and travel-time estimates that assist higher-level VRP algorithms and cost evaluation in subsequent stages.

Clarke-Wright Savings demonstrated excellent scalability for quick system-level routing, resulting in substantially condensed multi-stop routes with minimal runtime. The system created a limited collection of depot-based tours, each beginning and finishing at City_61, and covered clustered destinations via iterative route merging driven by distance "savings." The number of stops per tour showed significant consolidation compared to shortest-path approaches, which do not create multi-customer routes. CW also seemed to capture proximity-based grouping, with longer chains emerging after initial clusters were established. From an operational point of view, fewer routes can minimize dispatch workload, fleet division, and fixed vehicle expenses, even if single tours become longer than back and forth excursions. However, Clarke-Wright Savings's constructive merging creates rigidity because early merge decisions may lock in route architectures that are still possible but not globally optimal. Long tours show that one-pass greedy merging may be less cost efficient than metaheuristics that can iteratively alter assignments.

```
CW runtime: 0.64s

== CW Routes ==

  CW1 stops=7 dist(M)=2257407 cost=223483.29 truck=16.5 City_61 → City_19 → City_20 → City_16 → City_17 → City_18 → City_15 → City_51 → City_61

  CW2 stops=17 dist(M)=4234944 cost=698765.76 truck=16.5 City_61 → City_49 → City_22 → City_26 → City_36 → City_12 → City_10 → City_11 → City_30 → City_24 → City_28 → City_29 → City_31 → City_27 → City_25 → City_23 → City_8 → City_32 → City_61

  CW3 stops=22 dist(M)=7028661 cost=1897738.47 truck=16.5 City_61 → City_50 → City_39 → City_1 → City_2 → City_0 → City_37 → City_38 → City_33 → City_42 → City_47 → City_48 → City_45 → City_46 → City_40 → City_41 → City_6 → City_7 → City_57 → City_56 → City_58 → City_52 → City_21 → City_61

  CW4 stops=15 dist(M)=8389734 cost=1963197.76 truck=16.5 City_61 → City_43 → City_44 → City_59 → City_60 → City_3 → City_14 → City_13 → City_34 → City_35 → City_4 → City_55 → City_5 → City_53 → City_54 → City_9 → City_61
```

**FIGURE 3. Sample outputs from clarke-wright savings.**

The Greedy Multi-Vehicle heuristic generated feasible solutions quickly, demonstrating strong computational feasibility for time-sensitive planning. It builds routes incrementally by repeatedly selecting the next "best" viable destination given restrictions such as vehicle capacity and timing feasibility, and it ceases a route when no more destinations can be added. Greedy Multi-Vehicle's results revealed modest consolidation which is many routes resembled single-stop depot-return journeys, with only a few multi-stop routes. This pattern is consistent with greedy logic, locally attractive actions might result in early route termination, forcing remaining deliveries onto new vehicles. The considerable variety in route costs implies that decreasing the next-step distance does not always reduce system-level operating costs, especially when truck assignments vary between routes. As a result, Greedy Multi-Vehicle tends to increase fleet usage and division, raising costs such as dispatch effort and driver hours. Despite these shortcomings, Greedy Multi-Vehicle serves as a good benchmark for "fast feasibility" and shows the limitations of just local decision making.

The Genetic Algorithm took longer to compute than CW and Greedy Multi Vehicle due to its population-based search and iterative evaluation, but it delivered higher-quality fleet-level solutions. Genetic Algorithm outputs blended single-stop routes with selectively consolidated multi-stop routes, demonstrating that consolidation took place when it improved global goal values rather than when it was pushed uniformly. The presence of regular multi-stop chains indicates that Genetic Algorithm discovered favorable sequencing patterns under evolutionary pressure rather than by accident. By experimenting with different permutations and route partitions, Genetic Algorithm can avoid local optima that limit greedy techniques. Unlike Clarke Wright's one-pass merging, the Genetic Algorithm may improve previous decisions throughout generations, enhancing global configuration through selection, crossover, and mutation. Overall, the Genetic Algorithm outperformed the other methods in this study in terms of route quality and runtime, indicating that it is the best technique for constraint-driven optimization. The measured runtime is consistent with many practical planning cycles, particularly where solution quality leads to lower operating costs and better fleet utilization.

```
GA-MV runtime: 2.33s

== GA-MV Routes ==

GA48 stops=3 dist(M)=1720046 cost=5160.14 truck=16.5 City_61 → City_30 → City_22 → City_23 → City_61

GA2 stops=4 dist(M)=3281570 cost=9844.71 truck=16.5 City_61 → City_47 → City_46 → City_42 → City_41 → City_61

GA8 stops=4 dist(M)=4733680 cost=14201.04 truck=16.5 City_61 → City_29 → City_9 → City_11 → City_59 → City_61

GA4 stops=3 dist(M)=5443866 cost=16331.60 truck=16.5 City_61 → City_56 → City_60 → City_40 → City_61
```

**FIGURE 4. Snippets of genetic algorithm output.**

Ant Colony Optimization generated competitive route designs and proved effective multi-stop chaining, but it required the most computer resources due to frequent solution creation and pheromone updates. Ant Colony Optimization uses probabilistic construction and pheromone reinforcement to balance exploration and exploitation, promoting route segments that appear in high-quality solutions. The results show that Ant Colony Optimization may combine deliveries into coordinated multi-stop tours, frequently exceeding solely greedy selection by learning advantageous sequencing patterns across iterations. However, Ant Colony Optimization's performance is affected by parameter selections such as evaporation rate, pheromone weighting, number of ants, and iteration count. In this study, the higher runtime did not always explain the marginal advantages relative to Genetic Algorithms, which had a better runtime-to-quality trade-off. The comparative examination of all algorithms reveals a shift from path-level optimality to fleet-level optimization. Dijkstra and A* provide critical distance baselines; Clarke Wright provides quick consolidation; Greedy provides fast feasibility; and the genetic algorithm delivers the best overall scalable optimization under limitations.

| Algo | Num_Routes | Avg_Dist_per_Route (M) | Avg_Cost_per_Route | Cost_per_km_Overall | Fleet_Size | Truck_Types_Used | On_Time(%) |
|---|---|---|---|---|---|---|---|
| GA-MV | 51 | 2489415.02 | 44646.46875 | 17.93452212 | 432 | 2 | 99.16038751 |
| Greedy-MV | 53 | 2501975.434 | 42967.83611 | 17.17356435 | 440 | 2 | 98.72981701 |
| ACO-MV | 51 | 2502618.294 | 44686.07857 | 17.85573081 | 432 | 2 | 98.34230355 |
| CW | 4 | 5477686.5 | 1195796.32 | 218.3031686 | 256 | 1 | 8.719052745 |

**FIGURE 5. Comparison of the 4 algorithms.**

## IV. CONCLUSION

This study investigated how data-driven approaches might increase route efficiency in large-scale transportation systems while adhering to real-world operational restrictions. By combining traditional shortest-path algorithms with heuristic and metaheuristic vehicle routing approaches, the study demonstrated that effective optimization necessitates both accurate network representation and adaptive decision-making that can handle capacity, time-window, and cost restrictions. A single experimental architecture ensured consistent preprocessing, constraint management, vehicle assignment, and evaluation metrics across methods, allowing for systematic comparison and a better understanding of each method's strengths and limits.

Classical shortest-path algorithms (Dijkstra and A*) provided reliable network-level analysis and distance estimation, consistently delivering best pathways for individual origin-destination pairings. These findings confirmed their value as fundamental tools for building distance matrices and facilitating higher-level optimization. However, the findings revealed limits for real-world logistics operations which order-level shortest pathways do not account for shared vehicles, route consolidation, delivery deadlines, or fleet-level cost minimization. As a result, shortest-path algorithms are best suited as enablers inside larger routing schemes rather than complete end-to-end solutions [14].

Heuristic methods (Clarke Wright and Greedy) significantly outperformed traditional approaches by creating feasible multi-stop routes fast and at a cheap computational cost. Clarke-Wright achieved great consolidation through savings-based mergers, whereas Greedy generated quick, feasible solutions under tight deadlines. Nonetheless, both heuristics were bound by local decision-making, limiting their capacity to consistently achieve globally optimal configurations, especially as routing complexity increased [15]. These findings suggest that while heuristics are useful for quick baseline planning and initial solution development, they may underperform in large-scale, constraint-rich contexts where global trade-offs dominate.

Metaheuristic techniques (Genetic Algorithms and Ant Colony Optimization) had the highest adaptability and overall routing performance [12]. In particular, the Genetic Algorithm continuously achieved the optimum balance of distance reduction, operational cost efficiency, fleet utilization, and on-time performance, demonstrating the power of population-based global search under many constraints. The study indicates that no single strategy is universally optimal and performance is determined by network scale, constraint extent, and operational objectives. Future research should expand the framework to include real-time and dynamic data (for instance, traffic and weather), hybrid models combining metaheuristics and machine learning, evaluation on bigger national-scale datasets, and multi-objective optimization including sustainability criteria. Expanding to multi-depot, heterogeneous fleet, and stochastic demand scenarios would increase realism and application.

The current system is predicated on deterministic route conditions and static journey distances. As a result, the computational studies did not directly represent stochastic elements like traffic congestion, weather disruptions, road closures, and fluctuating service delays. These variables may change trip times, cause delivery delays, and impact vehicle availability in real-world transportation systems [11]. The model may be expanded in the future to include dynamic edge weights, which would alter trip expenses based on current traffic or weather. In a comparable way, stochastic node-based delays might be used to mimic different service times at consumer locations. This would make the framework more appropriate for real-time logistics planning by enabling the algorithm to carry out adaptive re-optimization when travel or delivery circumstances change.

## ACKNOWLEDGMENT

The authors wish to acknowledge the anonymous reviewers who made some very important comments and suggestions to help improve the quality and readability of this paper.

## FUNDING STATEMENT

There is no funding agencies supporting the research work.

## AUTHOR CONTRIBUTIONS

Ching Joe Tan: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Khai Wah Khaw: Project Administration, Writing – Review & Editing.

## CONFLICT OF INTERESTS

No conflict of interest was disclosed.

## ETHICS STATEMENTS

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

## DECLARATION OF AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

No artificial intelligence tools were used in the research or writing of this article.

## REFERENCES

[1] K. Baghizadeh, D. Zimon and L. Jum'a, "Modeling and Optimization Sustainable Forest Supply Chain Considering Discount in Transportation System and Supplier Selection under Uncertainty," *Forests*, vol. 12, no. 8, pp. 964, 2021. DOI: https://doi.org/10.3390/f12080964

[2] G. Sun and T. Li, "Optimizing Logistics in Forestry Supply Chains: A Vehicle Routing Problem Based on Carbon Emission Reduction," *Forests*, vol. 16, no. 1, pp. 62, 2025. DOI: https://doi.org/10.3390/f16010062

[3] V. Milin, T. Stanivuk, I. Skoko and T. Bulić, "Dijkstra and A* Algorithms for Algorithmic Optimization of Maritime Routes and Logistics of Offshore Wind Farms," *Journal of Marine Science and Engineering*, vol. 13, no. 10, pp. 1863, 2025. DOI: https://doi.org/10.3390/jmse13101863

[4] P. Silveira, Â.P. Teixeira and C.G. Soares, "AIS Based Shipping Routes Using the Dijkstra Algorithm," *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 13, no. 3, pp. 565-571, 2019. DOI: https://doi.org/10.12716/1001.13.03.11

[5] N. Tyagi, J. Singh and S. Singh, "A Review of Routing Algorithms for Intelligent Route Planning and Path Optimization in Road Navigation," *Lecture Notes in Mechanical Engineering*, pp. 851-860, 2023. DOI: https://doi.org/10.1007/978-981-19-4606-6_78

[6] J. Fikejz, M. Brázdová and Ľ. Jánošíková, "Modification of the Clarke and Wright Algorithm with a Dynamic Savings Matrix," *Journal of Advanced Transportation*, vol. 2024, no. 1, 2024. DOI: https://doi.org/10.1155/2024/8753106

[7] P. Borisovsky, "A parallel greedy approach enhanced by genetic algorithm for the stochastic rig routing problem," *Optimization Letters*, vol. 18, no. 1, pp. 235-255, 2024. DOI: https://doi.org/10.1007/s11590-023-01986-x

[8] A.K. Garside, L. Erlinda and I. Amallynda, "Solving heterogeneous fleet vehicle routing problem with clarke wright saving heuristic and genetic algorithm," *AIP Conference Proceedings*, vol. 3110, pp. 050002, 2024. DOI: https://doi.org/10.1063/5.0192216

[9] W. Li, L. Xia, Y. Huang and S. Mahmoodi, "An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 3, pp. 1557-1571, 2022. DOI: https://doi.org/10.1007/s12652-021-03120-0

[10] Y. Li, Z. Yang, S. Zhang and W. Liu, "A Study of the Capacitated Vehicle Routing Problem with Time-Window and Three-Dimensional Loading Constraints in Land–Sea Transport," *Sustainability*, vol. 16, no. 23, pp. 10272, 2024. DOI: https://doi.org/10.3390/su162310272

[11] Y. Xu and Y. Li, "A Real-Time Urban Traffic Congestion Prediction Framework Based on Dynamic Risk Field and Multi-Source Data Fusion," *IEEE Access*, vol. 13, pp. 172471-172495, 2025. DOI: https://doi.org/10.1109/access.2025.3608954

[12] 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. 13187-13257, 2023. DOI: https://doi.org/10.1007/s10462-023-10470-y

[13] A.P. P. and R. R., "Optimization of electric vehicle routing: A multi-criteria intelligent decision framework for efficient and scalable mobility," *Energy Reports*, vol. 15, pp. 108912, 2026. DOI: https://doi.org/10.1016/j.egyr.2025.12.049D.

[14] Rachmawati and L. Gustin, "Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem," *Journal of Physics: Conference Series*, vol. 1566, no. 1, pp. 012061, 2020. DOI: https://doi.org/10.1088/1742-6596/1566/1/012061

[15] J. Mašek, A. Pálková and Z. Bulková, "Application of the Clark–Wright Method to Improve the Sustainability of the Logistic Chain," *Applied Sciences*, vol. 14, no. 21, pp. 9908, 2024. DOI: https://doi.org/10.3390/app14219908