

Vehicle Routing Problem with Time Window for Disruption Management using Genetic Algorithm

Muhammad Amir Affan Mohamaden¹, Siti Suhana Jamaian^{1*}

¹ Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, UTHM Kampus Cawangan Pagoh, Hab Pendidikan Tinggi Pagoh, KM 1, Jalan Panchor, 84600, Pagoh, Muar, Johor, MALAYSIA.

*Corresponding Author: suhana@uthm.edu.my

DOI: <https://doi.org/10.30880/ekst.025.05.02.012>

Article Info

Received: 30 December 2024

Accepted: 18 January 2025

Available online: 19 December 2025

Keywords

Genetic Algorithm, Vehicle Routing Problem, Disruption Management, Optimization, Logistics

Abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is a critical logistics challenge that involves planning delivery routes to meet customer demands within specific time constraints while minimizing costs. Real-world disruptions, such as vehicle breakdowns and road blockages, can significantly affect delivery efficiency and operational costs. Addressing these challenges requires robust optimization methods. This study aims to solve the VRPTW by utilizing a Genetic Algorithm (GA) to optimize delivery routes under different disruption scenarios. The GA dynamically adjusted routes to handle disruptions while adhering to constraints such as vehicle capacity and time windows. The algorithm was implemented in Python, and three scenarios were analysed. The first scenario is a baseline scenario with no disruptions, where the total cost was RM652.62 while the second scenario is a road blockage scenario, where the GA rerouted deliveries and resulted in a cost of RM712.02 and the last scenario is a vehicle breakdown scenario, where tasks were reassigned to other vehicles, resulting in a cost of RM805.24. The results demonstrate the effectiveness of the GA in reducing costs, managing disruptions, and improving delivery performance. This research highlights the potential of optimization algorithms to enhance delivery operations, offering cost savings and improved efficiency in real-world logistics.

1. Introduction

The Vehicle Routing Problem with Time Windows (VRPTW) focuses on optimizing delivery routes to minimize costs and travel times while ensuring deliveries are completed within predefined time windows. Time windows are critical constraints in logistics, defining the earliest and latest times that vehicle can deliver to each customer. They are categorized into hard time windows, which require strict adherence, and soft time windows, which allow flexibility with associated penalties. Managing these constraints effectively is essential for maintaining customer satisfaction and operational efficiency.

Real-world scenarios, however, often face disruptions like traffic congestion or vehicle breakdowns, adding to the complexity of the problem. These disruptions can result in significant delays, increased operational costs, and reduced customer satisfaction. Genetic Algorithms (GA) offer an innovative approach by mimicking the natural selection process to adaptively optimize routes and improve resilience in the face of such challenges [1], [2], and [3]. GA enables dynamic rerouting, resource adjustments, and schedule updates to address disruptions, ensuring on-time deliveries and cost savings while maintaining high service levels [4].

This study develops optimal solutions for VRPTW under disruption scenarios using GA, proposing effective vehicle routing strategies for disruption management in logistics operations. It focuses on delivery routes for a

healthy beverage company based in Skudai, Johor, Malaysia, incorporating real-world constraints such as vehicle capacities and time windows [5], [6]. Python programming is used for algorithm development and simulations, leveraging its computational efficiency and flexibility for optimization tasks.

The research not only addresses theoretical challenges but also provides practical tools for managing disruptions in logistics. It offers a framework for real-time route optimization that can be applied to industries like e-commerce, supply chain management, and last-mile delivery services [7], [8], and [9]. By bridging the gap between academic research and real-world applications, this study contributes significantly to the field of logistics optimization, enhancing both operational resilience and cost efficiency.

2. Methodology

This research focuses on creating an optimization model to solve vehicle routing challenges in disruption management, specifically the Vehicle Routing Problem with Time Windows (VRPTW). Since VRPTW is considered a complex and time-consuming problem to solve manually, a Genetic Algorithm (GA) was used as the main solution method. The study was conducted within a logistics network to gather relevant data for the system's delivery and transportation needs. By using the GA, the project aimed to improve route efficiency and lower operational costs. Python programming was used to implement and fine-tune the algorithm, making it possible to address complicated routing and scheduling problems, even when unexpected disruptions like vehicle breakdowns or road blockages occurred.

2.1 Objective Function

The Vehicle Routing Problem with Time Windows involves planning optimal delivery routes from a central depot to various destinations while adhering to specific constraints. Each vehicle has a fixed capacity Q and services a group of customers within a directed weighted graph $G(V, A)$, where $V = \{v_0, v_1, \dots, v_n\}$ represents nodes (customers) starting from the depot v_0 , and $A = \{(v_i, v_j) | i, j = 0, 1, 2, \dots, n, i \neq j\}$ denotes arcs between locations [5], [8]. Here, i and j represent two nodes (locations) in the delivery network, where i refers to the starting location (depot or customer) and j refers to the destination location (another customer or the depot). Each customer i has a demand q_i , a service time s_i , and a time window $[a_i, b_i]$, where a_i is the earliest service start time and b_i is the latest. If the vehicle arrives at a customer before a_i , it must wait until service can begin. The arcs are associated with distances h_{ij} and travel times t_{ij} , representing the path between nodes i and j . The objective is to minimize the total cost F while ensuring deliveries meet all customer demands and respect their time windows [2]. This optimization problem is crucial for streamlining logistics operations, particularly for delivering perishable goods efficiently [1], [2]. The optimization process will continue iteratively until the solution converges, meaning the algorithm has reached a stable optimal value with minimal or no further improvement. In cases where the solution diverges, indicating unstable or unbounded results, the process will be terminated to prevent inefficiencies.

$$F = \sum_{e \in K} \sum_{(i,j) \in k} h_{ij} x_{ije} \quad (1)$$

where:

- F = Total cost
- h_{ij} = Distance from i to j
- e = Vehicle index from the set of available vehicles K
- x_{ije} = Equal to 1 if (i, j) is used by vehicle e and 0 otherwise
- K = Set of vehicles
- k = Node set $(0, 1, \dots, n + 1)$

2.2 Constraint

Equation (2) ensures that each customer j is visited exactly once [8].

$$\sum_{i \in N} x_{ij} = 1, \forall i \in N \quad (2)$$

where:

- x_{ij} = a binary variable indicating whether the arc (i, j) is used (1 if used, 0 otherwise)
- N = represents the set of customers in the problem

Equation (3) ensures that every vehicle departs from and returns to the depot (denoted as node 0) exactly once [4].

$$\sum_{j \in N} x_{0j} = 1 \quad \sum_{i \in N} x_{i0} = 1 \quad (3)$$

where:

- x_{0j} = a binary decision variable indicating whether a vehicle departs from the depot (node 0) to node j
- x_{i0} = a binary decision variable indicating whether a vehicle travels from node i to the depot (node 0)

Equation (4) ensure smooth operation with customers schedules, prevent missed deliveries [4]

$$e_j \leq t_j \leq l_j \quad (4)$$

where:

- e_j = Earliest allowable start time at the customer j
- l_j = Latest allowable start time at the customer j
- t_j = Actual service starts at customer j

Equation (5) ensures that delivery routes are planned such that no single vehicle is overloaded [3], [5].

$$\sum_{j \in N} q_j x_{ij} \leq Q, \forall i \in N \quad (5)$$

where:

- q_j = the demand at customer j
- Q = vehicle capacity.
- N = represents the set of customers in the problem

Equation (6) ensures punctuality and continuity of the routes [2], [8].

$$t_i + s_i + t_{ij} \leq t_j, \forall (i, j) \in A \quad (6)$$

where:

- t_i = start time of service at customer i
- s_i = service time at the customer i
- t_{ij} = travel time from customer i to j
- A = represents the set of arcs in the graph, where each arc is a pair (i, j)

Equation (7) ensures that no routes exceed a predefined time limit [4].

$$\sum_{(i,j) \in A} d_{ij} x_{ij} \leq D, \forall k \in K \quad (7)$$

where:

- d_{ij} = distance between customer i and j
- x_{ij} = a binary variable indicating whether the arc (i, j) is used (1 if used, 0 otherwise)
- K = Set of all vehicles
- D = represents the maximum allowable duration or length for a single vehicle's route.

Managing Vehicle Routing Problems with Time Windows means dealing with real-world challenges that make route planning both exciting and complex. For example, every customer needs to be visited exactly once, vehicles must start and finish at the depot, and all deliveries must happen within the specified time windows. On top of that, vehicles cannot carry more than their capacity, routes need to stay within reasonable time limits, and the schedule must flow seamlessly without any unnecessary detours. Tackling these challenges is not just about number; it is about creating smart, efficient routes that save time, cut costs, and keep everyone happy.

2.3 Genetic Algorithm

The implementation of Genetic Algorithm begins with the initialization of solutions as chromosomes. These chromosomes represent potential delivery routes, encoded to reflect both route sequences and constraints [10]. The algorithm proceeds as follows:

- Step 1 : Initialization: Generate an initial population of chromosomes, each representing a feasible route. Randomization and heuristic methods were employed to ensure diversity in the initial population.
- Step 2 : Selection: Perform roulette-wheel selection to prioritize chromosomes with higher fitness scores. Fitness is evaluated based on the total cost, distance, and adherence to time windows.

- Step 3 : Crossover: Apply single-point or multi-point crossover operations to combine parent chromosomes, generating offspring with traits from both parents. This promotes genetic diversity and explores new solution spaces.
- Step 4 : Mutation: Introduce variability through mutation operators such as insertion, inversion, and swapping. These mutations address local optima challenges and ensure exploration of the solution space.
- Step 5 : Evaluation: Assess the fitness of offspring chromosomes based on their cost, distance, and constraint violations. Infeasible solutions are penalized to maintain solution quality.
- Step 6 : Termination: Repeat the process for a fixed number of generations or until convergence criteria, such as minimal improvement in fitness, are met. The best-performing chromosome is selected as the final solution.

2.4 Related Data

The information for this study was obtained from the logistics department of a healthy beverage company, based in Skudai, Johor, Malaysia. Location of the customer, customer demand and service time were collected. Python software was used to implement the main algorithm, the Genetic Algorithm (GA). The company distributes their product directly to the customer.

The warehouse of the company is located at Skudai, while the locations of the customers are included in Larkin, Bandar Baru Uda, Stulang, Kampung Melayu Majidee, Taman Suria, Tebrau and Kempas. All locations are in South Peninsular Malaysia. The delivery for the customer will follow the trip that has been organized. Fig. 1 shows the location of the delivery.

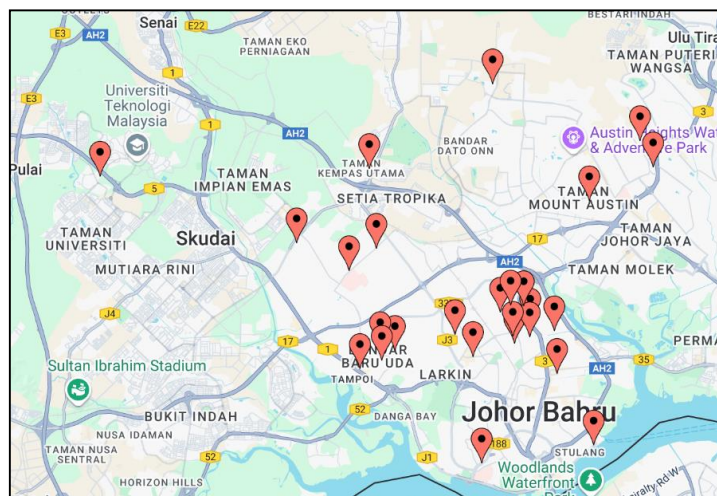


Fig. 1 Location of the Customers

Table 1 shows the location for each node that delivers the healthy beverage product. Node 0 is the depot or the warehouse where the delivery will start from the warehouse and go back to the warehouse after the delivery.

Table 1 Location for each customer

Node	Location
0	Jalan Pontian Lama
1	Jalan Masjid
2	Jalan Padi Emas 3/1
3	Jalan Harmonium 10
4	Jalan Seri Austin 3/14
5	Jalan Garuda
6	Jalan Suria 3
7	Jalan Anggerik 1
8	Jalan Larkin Jaya 4
9	Jalan Sutera 2
10	Jalan Teratai

11	Jalan Suria Permai 1
12	Jalan Tebrau Lama
13	Jalan Padi Mahsuri
14	Jalan Ibrahim Sultan
15	Jalan Padi Malinja 4
16	Jalan Kempas Utama 2/4
17	Jalan Suria Utama
18	Jalan Cengkih 3
19	Jalan Mawar
20	Jalan Kempas Baru 4
21	Jalan Padi Ria 2
22	Jalan Harmonium 5
23	Jalan Lingkaran Dalam
24	Jalan Persiaran Kempas Baru
25	Jalan Austin Perdana 2

This research is focused on the routes of delivery for disruption management in the healthy beverage industry company to customers in South Peninsular Malaysia. There are one warehouse and one customer or node for every route which is to specify that Route 1 has 3 locations of customers, Route 2 has 3 locations of customers, Route 3 has 3 locations of customers, Route 4 has 4 locations of customers, Route 5 has 4 locations of customers, Route 6 has 3 locations of customers, Route 7 has 2 locations of customers and Route 8 has 3 locations of customers. Table 2 through Table 9 shows the detail for each route.

Table 2 Data for Route 1

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
1	12	151	171	20
2	8	40	60	20
3	12	106	126	20

Table 3 Data for Route 2

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
4	12	139	159	20
5	6	24	44	20
6	6	89	109	20

Table 4 Data for Route 3

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
7	12	71	91	20
8	8	85	105	20
9	10	87	107	20

Table 5 Data for Route 4

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
10	18	114	134	20
11	12	57	77	20
12	12	53	73	20
13	10	149	169	20

Table 6 Data for Route 5

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
14	8	22	42	20
15	6	51	71	20
16	12	65	85	20
17	12	147	167	20

Table 7 Data for Route 6

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
18	12	77	97	20
19	18	66	86	20
20	8	116	136	20

Table 8 Data for Route 7

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
21	12	52	72	20
22	6	87	107	20

Table 9 Data for Route 8

Node	Demand per Product	Ready Time (minute)	Due Time (minute)	Service Time (minute)
0	0	0	230	0
23	8	58	78	20
24	8	143	163	20
25	12	162	182	20

In this dataset, the start time for Node 0 (depot) is always set to 0, and the due time is 230 minutes. This is because the company has established a policy that each route must be completed within a maximum of 230. This ensures timely delivery and efficient routes management. For all customer nodes, the service time is fixed at 20 minutes. This standardization aligns with the company's operational requirement that each customer must be served within 20 minutes. This includes time for unloading goods, completing paperwork, or addressing customer-specific needs. Each customer node has unique ready times and due times, which reflect differences in travel distances from the depot or between consecutive customer nodes. These variations account for the travel time required to reach each location while ensuring that routes comply with the company's 230-minute operational window. The combination of standardized service times and varying travel times forms the foundation for optimizing routes and evaluating the impact of the GA.

Before the optimization process, the total cost for the delivery routes was calculated based on a total travel distance of 1458.35, resulting in an expense of RM729.17. This calculation reflects the baseline cost of operating the routes without any optimization applied, serving as the initial reference point for evaluating the effectiveness of the GA.

3. Results and Discussion

3.1 Post-Optimization

The total cost after the optimization, calculated without considering any penalties for disruptions, amounted to RM652.62, reflecting the reduced expense achieved through the optimization process. The optimization process reached its best result at iteration 1091 because this iteration provided the lowest total cost while satisfying all constraints, such as time windows and vehicle capacity. This indicates that the algorithm had refined the solution to a point where further iterations showed minimal or no improvement, marking convergence to an optimal solution. The fitness value of 122.31, chosen as the best fitness, represents the algorithm's evaluation of the solution's quality, combining factors like total cost, adherence to constraints, and efficiency of the routes. This fitness score ensures a balance between minimizing costs and maintaining operational feasibility. Although the total travel distance increased to 1631.56 (a difference of 173.21, representing an 11.88% increase), the optimization successfully reduced the total cost by RM76.55, achieving a cost improvement of 2.62%. The increase in travel distance occurred because the number of routes increased from 8 to 11. This was due to the shorter distances between certain nodes, which allowed the Genetic Algorithm (GA) to split routes more effectively to minimize costs. By creating additional routes with shorter distances between nodes, the algorithm was able to optimize vehicle utilization and reduce overall operational expenses. These results demonstrate the efficiency of the GA in optimizing route configurations, showcasing its capability to balance operational expenses, route adjustments, and effective resource utilization.

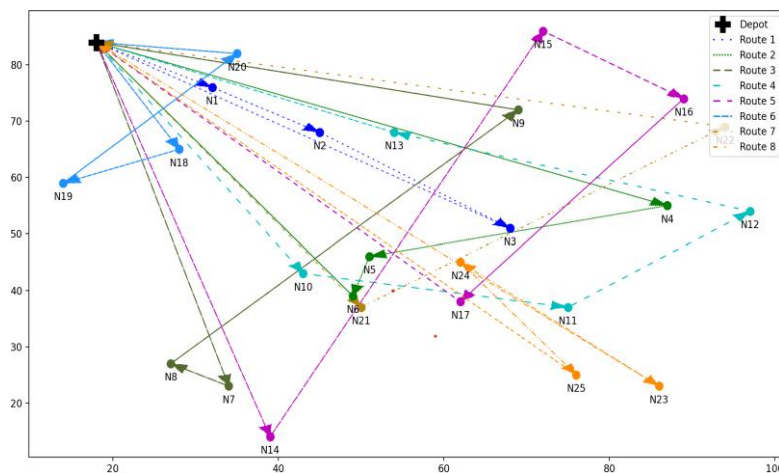


Fig. 2 The route after the optimization

This graph represents the optimized delivery routes for a Vehicle Routing Problem with Time Windows (VRPTW) scenario. Each route is color-coded and connects the depot (marked as a black cross) to customer nodes (labelled as N1, N2, ..., N25), ensuring all constraints like time windows and vehicle capacity are satisfied. The optimization increased the number of routes from 8 to 11, as shown by the distinct paths, due to shorter distances between some nodes, which allowed for more efficient clustering of deliveries. Despite the increase in total travel distance, this configuration minimized overall operational costs, showcasing the effectiveness of the GA in balancing travel distances, resource utilization, and cost efficiency.

3.2 Post-Optimization with Blockage

In this Genetic Algorithm (GA), blockages are managed dynamically by identifying specific customer locations where access is temporarily restricted. When a blockage occurs, the program applies a penalty cost for each affected customer and adjusts the delivery routes to bypass the blocked locations while maintaining efficiency and adhering to constraints like vehicle capacity and time windows. Each blockage incurs a penalty of RM100, which is added to the total cost. For example, in the case of blockages occurring at customer locations [1, 3, 5, 7, 9, 11], the program successfully rerouted the delivery operations to avoid these blocked nodes without surpassing the predefined cost threshold of RM2000.

The total cost before the optimization process, calculated based on a total travel distance of 1458.35, amounted to RM729.17, reflecting the baseline expense for the delivery routes. However, due to blockages at nodes 1, 3, 5, 7, 9, and 11, an additional penalty was incurred, increasing the total cost to RM1329.17. This calculation highlights the inefficiencies and increased expenses caused by the disruptions in the original routes. Thus, the optimization process aims to address these challenges by reducing the total cost and improving delivery efficiency, despite the presence of blockages.

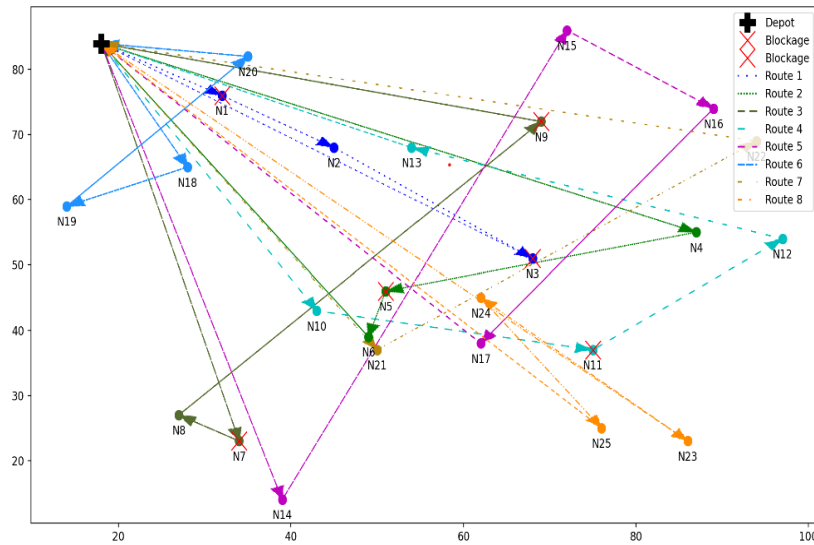


Fig. 3 The route before the optimization with blockage

Fig. 3 shows the delivery routes before the optimization with blockages at locations 1, 3, 5, 7, 9, and 11, marked with red crosses. The depot is represented by the black cross, and each route is shown in a different colour. The GA rerouted around the blocked locations while ensuring deliveries to other nodes were completed efficiently and within time and capacity constraints.

The total cost after the optimization, including penalties for blockages at nodes 1, 3, 5, 7, 9, and 11, amounted to RM1312.02. The optimized travel distance was 1780.04, reflecting an increase of 321.69 compared to the original distance. This represents a 22.06% decrease in distance efficiency, primarily due to the increase in the number of routes from 8 to 11. The additional routes allowed the GA to reroute around blockages more effectively, ensuring adherence to constraints like time windows and vehicle capacity while minimizing the impact of disruptions. Despite the increased travel distance, the optimization successfully reduced the total cost by RM17.16, achieving a slight improvement of 0.59% compared to the original cost of RM1329.17. This improvement demonstrates the GA's ability to prioritize cost reduction over distance efficiency, effectively balancing operational expenses and adapting to challenging conditions caused by multiple blockages.

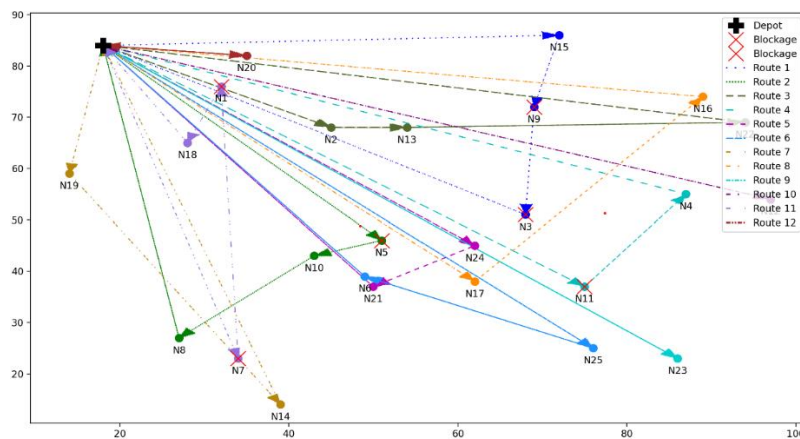


Fig. 4 The route after the optimization with blockage

Fig. 4 shows the optimized delivery routes with blockages at locations 1, 3, 5, 7, 9, and 11, marked by red crosses. The depot is represented by the black cross, and each route is displayed in a different colour. To adapt to

the blockages, the GA added more routes, increasing the number of routes from 8 to 12. This adjustment allowed the algorithm to reroute deliveries efficiently while avoiding blocked nodes and ensuring all constraints, such as time windows and vehicle capacity, were met.

3.3 Post-Optimization with Vehicle Breakdown

In this Genetic Algorithm (GA), vehicle breakdowns are managed by dynamically adjusting the delivery routes to minimize their impact. The system is designed to simulate scenarios where a single vehicle breakdown can occur during the delivery process. When a breakdown is specified, the program identifies the affected vehicle and redistributes its assigned deliveries to other operational vehicles. This ensures that customer demands are met without significant delays or disruptions.

The total cost before the optimization process, calculated based on a total travel distance of 1458.35, amounted to RM729.17, reflecting the baseline expense for the delivery routes. However, due to a vehicle breakdown at node 9, a penalty of RM150.00 was incurred, increasing the total cost to RM879.17. Before the breakdown, eight vehicles were assigned, each covering a single route. After the breakdown occurred, there were still eight routes because these represent the original routes before optimization, meaning no rerouting had yet taken place to adjust for the breakdown. The lack of rerouting initially resulted in inefficiencies, as the remaining seven vehicles had to handle the deliveries of the broken-down vehicle.

The best result was found at iteration 231, with a fitness value of 109.97. Iteration 231 was chosen because it marked the point where the algorithm achieved the lowest cost while maintaining feasible delivery routes and satisfying all constraints, including vehicle capacity and time windows. The fitness value of 109.97 reflects the quality of the solution, balancing the total cost and operational efficiency under the disrupted conditions. Thus, this demonstrates the GA's ability to adapt, minimize costs, and reassign deliveries effectively, ensuring optimal delivery performance despite the reduced number of available vehicles.

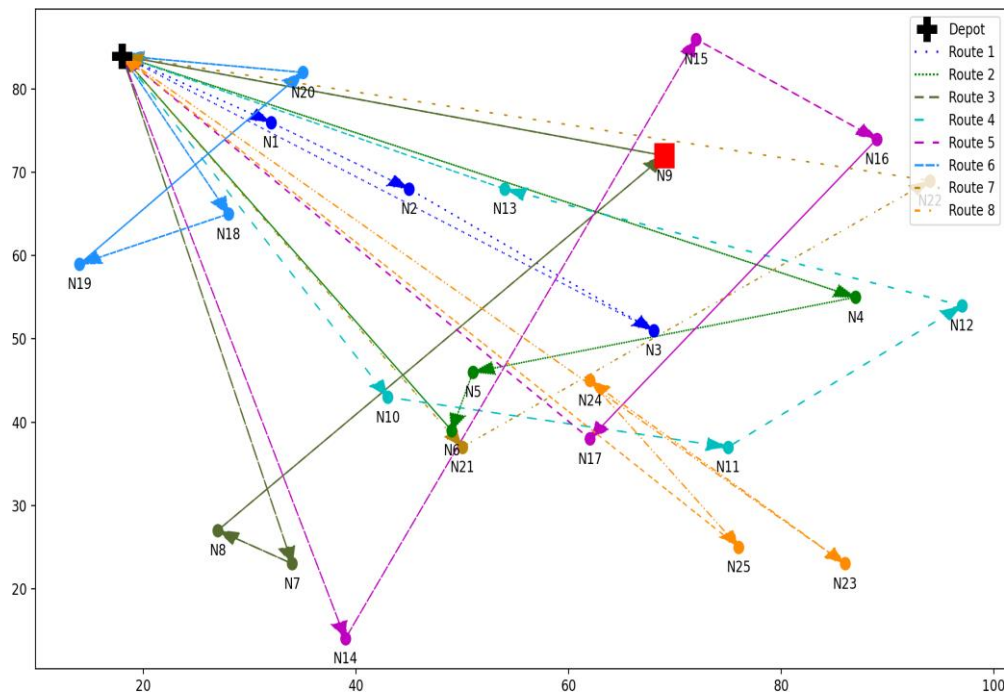


Fig. 5 The route before the optimization with vehicle breakdown

Fig. 5 illustrates the delivery routes before the optimization, with a vehicle breakdown occurring at node 9, marked by the red square. The depot is represented by the black cross, and each route is color-coded. This setup highlights the initial impact of the disruption, where the routes were still based on the original configuration without adjustments for the breakdown.

The total cost after the optimization, including penalties for a vehicle breakdown at node 9, amounted to RM805.24. The optimized travel distance was 1638.09, reflecting a reduction of 179.74 km compared to the original distance, resulting in a 12.32% increase in distance efficiency. This improvement occurred because the Genetic Algorithm optimized the delivery routes to better utilize the reduced number of vehicles by redistributing deliveries and creating more direct paths between customer nodes. While the distance efficiency improved, the algorithm also successfully reduced the total cost by RM73.94, achieving a cost improvement of 2.54% compared to the original cost of RM879.17. This reduction was achieved by prioritizing cost-effective routes and minimizing

penalties, even with fewer vehicles available. These results highlight the Genetic Algorithm's ability to adapt to disruptions caused by the vehicle breakdown, reassign deliveries among the operational vehicles, and optimize route configurations to balance both travel efficiency and operational costs under challenging conditions.

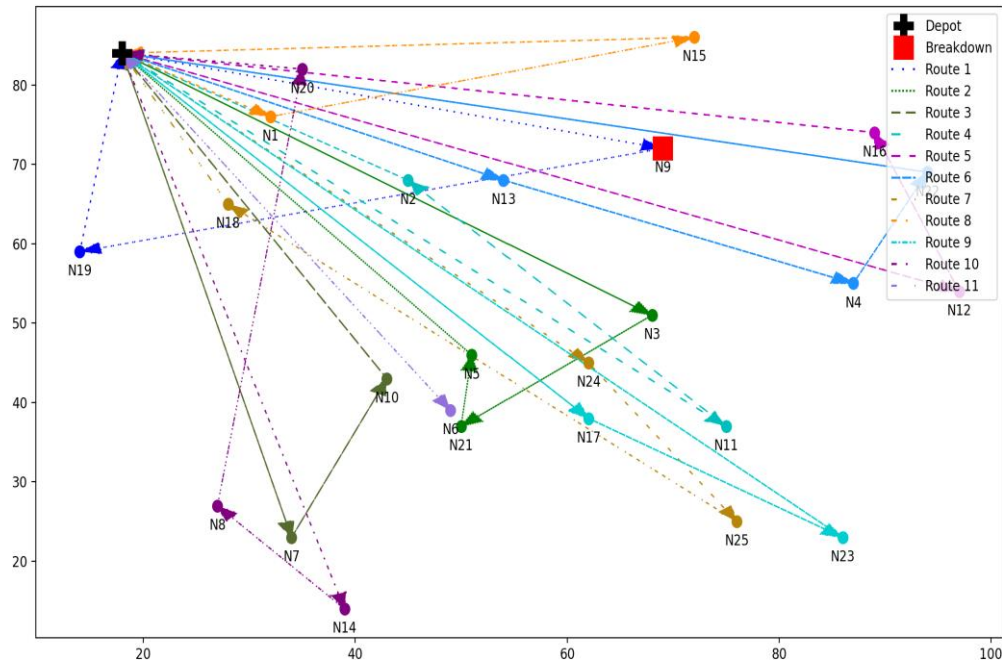


Fig. 6 The route after the optimization with vehicle breakdown

Fig. 6 illustrates the delivery routes after optimization, with the vehicle breakdown at node 9 still marked by the red square. The depot is represented by the black cross, and the routes are color-coded. Although the breakdown remains marked at node 9, it indicates that the disruption occurred before the optimization process. The GA successfully redistributed the deliveries originally assigned to the affected vehicle among the remaining operational vehicles, increasing the total number of routes to 11. This adjustment ensured all deliveries were completed efficiently while adhering to time and capacity constraints, demonstrating the algorithm's ability to adapt to disruptions and optimize delivery performance.

3.4 Impact of Disruption

Optimized routes successfully minimized the cumulative impact of six disruptions, maintaining total costs under RM2000. The dynamic adjustment of routes ensured that service levels were upheld, even in the face of substantial challenges. For example, when blockages occurred at multiple customer locations, the Genetic Algorithm (GA) rerouted deliveries efficiently, avoiding penalty escalation and ensuring the timely fulfilment of demands. This adaptability underscores the GA's robustness in handling complex, real-world scenarios.

4. Conclusion

Efficient route planning and cost reduction are key to solving the Vehicle Routing Problem with Time Windows (VRPTW). This study used a Genetic Algorithm (GA) implemented in Python to optimize delivery routes for a healthy beverage logistics network. The GA optimized routes for 25 customer nodes, considering vehicle capacity and time constraints, and improved them through operations like selection, crossover, and mutation. The program was run five times per scenario to ensure consistent and reliable results.

The results were verified by comparing the optimized costs and distances with the baseline values before optimization. The GA was also tested in scenarios with disruptions, like blockages and vehicle breakdowns, to confirm its adaptability. The consistent improvements in cost reduction, resource use, and route efficiency show that the GA effectively solved the problem and optimized delivery strategies.

Acknowledgement

The authors would also like to thank the Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Muhammad Amir Affan Mohamaden, Siti Suhana Jamaian; **validation of results:** Muhammad Amir Affan Mohamaden, Siti Suhana Jamaian; **data collection:** Muhammad Amir Affan Mohamaden; **analysis and interpretation of results:** Muhammad Amir Affan Mohamaden; **draft manuscript preparation:** Muhammad Amir Affan Mohamaden, Siti Suhana Jamaian. All authors reviewed the results and approved the final version of the manuscript.

References

- [1] Y. Huang and L. Xu, "Dynamic VRPTW models," *Transportation Science*, vol. 57, no. 2, 2023. DOI: 10.1287/trsc.2022.1179
- [2] R. Khoidir and J. Garside, "Adaptive routing under disruptions," *Journal of Logistics*, vol. 34, no. 3, 2022. DOI: 10.1080/13675567.2022.2024180
- [3] T. Bulhões, et al., "Stochastic approaches to VRPTW," *Operational Research*, vol. 45, no. 4, 2018. DOI: 10.1007/s12351-018-0395-7
- [4] N. El-Sherbeny, "Metaheuristic techniques for VRP," *Journal of Applied Operational Research*, vol. 2, no. 1, pp. 34–45, 2010. DOI: 10.13033/jaor.2020.2352
- [5] P. Toth and D. Vigo, *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics, vol. 10, no. 2, 2023. DOI: 10.1137/1.9780898718481
- [6] G. Laporte and I. H. Osman, "Routing problems: A bibliography," *Annals of Operations Research*, vol. 61, 1995. DOI: 10.1007/BF02032089
- [7] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, 1964. DOI: 10.1287/opre.12.4.568
- [8] B. Golden, S. Raghavan, and E. A. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer Optimization and Its Applications, vol. 43, 2008. DOI: 10.1007/978-0-387-77778-8
- [9] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, 1959. DOI: 10.1287/mnsc.6.1.80
- [10] A. Maroof, B. Ayvaz, and K. Naeem, "Logistics optimization using hybrid genetic algorithm (HGA): A solution to the vehicle routing problem with time windows (VRPTW), vol. 11, , 2023. DOI:10.1109/ACCESS.2023.0322000.