ENHANCED KNOWLEDGE IN SCIENCES AND TECHNOLOGY
e-ISSN: 2773-6385

EKST

Vol. 5 No. 2 (2025) 101-111
https://publisher.uthm.edu.my/periodicals/index.php/ekst

Vehicle Routing and Scheduling Problems in Supply Chain Management using Ant Colony Algorithm

Nur Aqilah Nadirah Che Amran[1], 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.2025.05.02.013

Article Info

Received: 30 December 2024
Accepted: 18 January 2025
Available online: 10 December 2025

Keywords

Supply Chain Management, Optimal Delivery Routes, Ant Colony Optimization

Abstract

Supply chain management plays a crucial role in ensuring smooth business operations and meeting customer demands effectively. Failure to optimize delivery scheduling and route selection can lead to inefficiencies, such as increased travel distances and operational challenges. This study addresses the Vehicle Routing and Scheduling Problem (VRSP) within the context of supply chain management, focusing on minimizing total travel distance. The research employs the Ant Colony algorithm, a powerful method for solving complex and nondeterministic polynomial-time (NP-hard) problems, implemented using Python software. Key factors considered include vehicle capacity, customer demand, and delivery time constraints, ensuring efficient route planning and delivery schedules. The study involves deliveries from a warehouse in Guar Chempedak, Kedah, to 19 customer locations across Kedah, such as Kulim, Kodiang, Jitra, Kampung Bukit Selambau, Pokok Sena, Sik, Amanjaya Sungai Petani, and others. Results reveal optimized travel distances for five delivery trips: 215.25 km for Trip 1, 167.01 km for Trip 2, 115.45 km for Trip 3, 113.04 km for Trip 4, and 181.56 km for Trip 5. These findings demonstrate the effectiveness of using the Ant Colony algorithm in selecting optimal delivery routes and achieving significant reductions in transportation distance. In conclusion, the study highlights the importance of optimizing route planning and scheduling in supply chain management to reduce travel distances. Although potential limitations, such as dynamic traffic conditions or unexpected delays, may influence real-world implementation, the outcomes provide valuable insights for enhancing delivery efficiency and improving overall supply chain performance.

1. Introduction

Vehicle routing and scheduling in supply chain operations are crucial for ensuring efficiency and minimizing costs. Over the past decade, supply chain management (SCM) has garnered significant attention and evolved into a widely researched domain. For instance, the term "supply chain" featured prominently in conferences, with its usage increasing from 13.5% in 1995 to 22.4% in 1997 at the Council of Logistics Management's Annual Conference [1]. SCM's growing importance is evident across industries such as manufacturing, distribution, and transportation, where the focus is on optimizing processes to reduce environmental impact and transportation expenses [2].

Vehicle Routing and Scheduling Problems (VRSP) are among the most challenging aspects of SCM, requiring efficient allocation of resources and optimization of routes to enhance operational effectiveness. The VRSP includes vehicle routing problems (VRP), where the goal is to determine optimal routes for a fleet to serve customers, and vehicle scheduling problems, which focus on the sequence of visits to minimize costs. Traditional methods like linear programming and heuristic algorithms have been extensively used to address these challenges [3].

Heuristic and metaheuristic approaches, including Genetic Algorithms, Simulated Annealing, and Ant Colony Optimization (ACO), play a pivotal role in solving VRSP. ACO, inspired by the foraging behaviour of ants, is particularly effective in optimization tasks. The ACO algorithm, a metaheuristic approach inspired by ant behaviour, has been proven effective in solving problems like the Traveling Salesperson Problem (TSP) and VRP. The algorithm uses pheromone trails and probabilistic decision-making to find efficient solutions iteratively [4]. Numerous studies have highlighted its success in handling VRP variants, such as capacitated VRP with time windows and VRP with simultaneous pickup and delivery [5]. Additionally, combining ACO with hybrid techniques has further improved its scalability and performance [6]. Accurate distance calculation is essential for route optimization in VRSP. The Haversine formula, which computes the shortest distance between two points on a spherical surface, is used to enhance precision in this study. Integrating the formula with ACO allows for more realistic modelling of delivery routes, contributing to effective planning and scheduling [7].

Thus, this study focuses on applying the ACO algorithm to vehicle routing and scheduling within SCM, with data sourced from the Food Industry in Guar Chempedak, Kedah. Python programming is employed to execute the VRP with a time windows approach, aiming to identify the most efficient routes based on minimum costs and optimal travel distances. By leveraging ACO and Python, this research seeks to optimize supply chain vehicle routes and schedules while proposing the best delivery routes for efficient and cost-effective supply chain management.

2. Methodology

This study focuses on developing a mathematical model to address scheduling and vehicle routing challenges within supply chain management. As the Vehicle Routing Problem is classified as a nondeterministic polynomial-time (NP-hard) problem, the Ant Colony algorithm is employed as the solution method. Integrating the Ant Colony algorithm is essential to enhance supply chain efficiency and streamline transportation operations within the industry. Python programming is utilized to implement and optimize the algorithm effectively, enabling the resolution of complex scheduling and routing tasks. The research is conducted within a production industry in Kedah to gather data tailored to its transportation requirements.

2.1 Objective Function

This formulation aims to minimize transportation costs while ensuring punctual delivery of goods to customers and preventing any vehicle from exceeding its capacity in supply chain management. Achieving this involves efficiently assigning customers to vehicles and determining the optimal sequence for their visits. The scheduling problem involves determining the optimal sequence and timing of activities, considering factors such as task durations, resource availability, precedence relationships, and any other constraints specific to the problem domain [8].

Various factors such as distance, time, vehicle capacity, and other specific constraints are considered during this process. Vehicle routing problem can be modelled as a directed weighted graph *G (V, E)* where $V = \{v_0, v_1, \dots, v_n\}$ represents the whole set of vertices and $E = \left[\{v_i, v_j\}, (i,j) = 0,1,2\, n,\ \ i \neq j\right]$ represents the set of arcs between the vertices. The depot is assigned as and v_0 the others v_j represent the customer. Each client has different demand, and the vehicle has a different capacity. The product must be delivered within the time that the customer wants and delivered to every customer at different times. Thus, the objective function can be mathematically represented as follows [9];

$$\text{Minimize Z} = \sum_{k \in K} \sum_{(i,j) \in A} C_{ij} X_{ijk} \tag{1}$$

Subject to

$$\sum_{k \in K} \sum_{j \in \Delta+1} X_{ijk} = 1 \qquad ;\forall i \in N \tag{2}$$

$$\sum_{j \in \Delta-\ (0)} X_{ijk} = 1 \qquad ;\forall i \in N \tag{3}$$

where:

K = set of vehicles with the same capacity

A = Node set = {0, 1, ..., n+1}
N = Customer set = {1, ..., n}
C_{ij} = Distance from node i and j
$\boldsymbol{x_{ijk}}$ = Equal to 1 if node *(i,j)* is used by vehicle k and 0 otherwise

Constraints

Let,

$\boldsymbol{w_{ik}}$ = start of service at node i when serviced by vehicle k
$\boldsymbol{t_{ij}}$ = time travel from node i to node j
$\boldsymbol{s_i}$ = service time node i
d_i = customer demand quantity i
E = earliest possible departure from the depot
L = latest possible arrival at the depot
$[\boldsymbol{\alpha_i}, \boldsymbol{b_i}]$ = time windows from node i to node j

Equation (4) defines vehicles arrival at customer locations within specified time windows and depart within a maximum allowable time.

$$\sum_{i\in\Delta-j} x_{ijk} - \sum_{i\in\Delta+(j)} x_{ijk} = 0 \qquad ;\forall k \in K, j \in N \tag{4}$$

Equation (5) ensures that vehicles depart from the depot, accounting for the absence of the vehicle at the depot. A time window $[\alpha_i , b_i]$ is defined where α_i represents the earliest possible service start time, and b_i indicates the latest. If a vehicle reaches a customer before $\boldsymbol{\alpha_i}$, it must remain idle until service is permitted to commence.

$$\begin{aligned} &\sum_{i\in\Delta-(n+1)} x_{i,n+1,k} = 1 && ;\forall k \in K \\ &x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \le 0 && ;\forall k \in K, (i,j) \in A \\ &a_i \sum_{j\in\Delta+(i)} x_{ijk} \le w_{ik} \le b_i \sum_{j\in\Delta+(i)} x_{ijk} && ;\forall k \in K, (i,j) \in A \end{aligned} \tag{5}$$

Equation (6) ensures the continuity of vehicle routes by specifying the arrival time at each customer relative to the arrival time at the previous customer.

$$\begin{aligned} &E \le w_{ik} \le L && ;\forall k \in K, i \in \{0, n+1\}, \\ &\sum_{i\in N} \sum_{j\in\Delta+(i)} x_{ijk} \le C && ;\forall k \in K \\ &\sum_{i\in N} d_i \sum_{j\in\Delta+(i)} x_{ijk} && ;\forall k \in K \end{aligned} \tag{6}$$

Equation (7) governs the assignment of customers to vehicles and the utilization of vehicles

$$\begin{aligned} &x_{ijk} \ge 0 && ;\forall k \in K, (i,j) \in A, \\ &x_{ijk} \in \{0,1\} && ;\forall k \in K, (i,j) \in A, \end{aligned} \tag{7}$$

In the context of Supply Chain Management, the Vehicle Routing Problem becomes increasingly complex as it involves integrating various real-world constraints. These constraints include things like differentiable vehicle capacity, route length restrictions, time slots for arrival and departure at every customer location, different service durations, and the need to pick up or deliver items. By leveraging Ant Colony algorithms, our research aims to address these challenges by optimizing vehicle routes and schedules to minimize the total distance.

2.2 Ant Colony Algorithms

The Ant Colony algorithm has been selected to solve this variant of the Vehicle Routing Problem (VRP). The behaviour of real ants in searching for food can be related to this VRP. The concentration of pheromones left by ants on a path assists other ants in determining whether to follow that path or not. This analogy to the behaviour of real ants aids the colony in optimizing their paths to efficiently transport food back to their nest.
The Ant Colony algorithm can be implemented as follows [10]:

Step 1: Start by initializing algorithm parameters such as the number of ants (k), the number of cycles (1), the pheromone decay rate (ρ), the initial pheromone level ($\boldsymbol{\tau_0}$), For each ant k probability $\boldsymbol{P_{k\,(i,j)}}$ of moving from the current node to another node or destination is calculated taking to the formula where $\boldsymbol{\alpha}$ ***and*** $\boldsymbol{\beta}$

are respectively parameters. Also, initialize the pheromone matrix (*N***N*), where each $\boldsymbol{\tau_{ij}}$ represents the level.

Step 2: Place ants at their starting points.

Step 3: Introduce a variable q, randomly sampled from a uniform distribution [0,1], to decide the next destination. If $q \leq \boldsymbol{q_0}$ (a predefined threshold), equation (8) is used for selection; otherwise, equation (9) is applied. The value of $\boldsymbol{q_0}$ controls the balance between exploitation and exploration.

Where $\boldsymbol{P_{k\,(i,j)}}$ is the probability of ant choosing customer (node) *j* after visiting customer (node) i on a route. $\boldsymbol{\tau_{ij}}$ is the pheromone of the distance of customer *i* to customer *j*, and $\boldsymbol{\eta_{ij}}$ is the visibility of the distance of customer *i* to customer *j* while $\boldsymbol{\eta_{iz}}$ represents the visibility of the path between node $\boldsymbol{i}$ and node *z*. α and β are, respectively the ant trail intensity controller where α is the relative importance of trail, and β is the relative importance visibility controller of the distance of customer i to customer j with values $\alpha \geq 0$ and $\beta \geq 0$. $\boldsymbol{j_k(i)}$ is the set of unvisited nodes for ant k after visiting node $\boldsymbol{i}$ and $\boldsymbol{\tau_{iz}}$ refers to the pheromone level on the path between node $\boldsymbol{i}$ and node *z*. $\boldsymbol{s}$ represents the current node (or stop) being considered in the route.

$$\boldsymbol{P_{k\,(i,j)}} = \frac{(\boldsymbol{\tau_{ij}})^{\boldsymbol{\alpha}}.(\boldsymbol{\eta_{ij}})^{\boldsymbol{\beta}}}{\sum(\boldsymbol{\tau_{ij}})^{\boldsymbol{\alpha}}.(\boldsymbol{\eta_{ij}})^{\boldsymbol{\beta}}}$$

$$\boldsymbol{P_{k\,(i,j)}} = \boldsymbol{arg} - \mathbf{max}\{(\boldsymbol{\tau_{iz}}).(\boldsymbol{\eta_{iz}})^{\boldsymbol{\beta}}\} \qquad \mathbf{(8)}$$

$$\boldsymbol{P_{k\,(i,j)}} = \left\{\frac{(\boldsymbol{\tau_{ij}}).(\boldsymbol{\eta_{ij}})^{\boldsymbol{\beta}}}{\sum_{z\in \boldsymbol{j_k}}(\boldsymbol{i})(\boldsymbol{\tau_{iz}}).(\boldsymbol{\eta_{iz}})^{\boldsymbol{\beta}}}\right\} \boldsymbol{if\ s} \in \boldsymbol{j_k(i)} \qquad \mathbf{(9)}$$

Step 4: If the ant has not visited all stops, repeat step 3. Once all stops are visited, return to the starting point and compute the total distance travelled. Update the local pheromone level using equation (10):

$$\boldsymbol{\tau_{ij}} = (\mathbf{1} - \boldsymbol{\rho}) * \boldsymbol{\tau_{ij}} + \boldsymbol{\rho} * \boldsymbol{\tau_{0,}} \qquad \mathbf{(10)}$$

Step 5: After all ants complete their tours in a cycle, determine the optimal route and update the global pheromone level using equation (11):

$$\boldsymbol{\tau_{ij}} = (\mathbf{1} - \boldsymbol{\rho}) * \boldsymbol{\tau_{ij}} + \boldsymbol{\rho}(\mathbf{1}_{\boldsymbol{mc}})^{-\mathbf{1}} \qquad \mathbf{(11)}$$

Step 6: Repeat steps 2-5 until the number of iterations reaches the specified limit from step 1. Finally, the algorithm returns to the best optimal path. Simplifies to $\boldsymbol{mc}$ representing the cost of the best solution.

For this algorithm, Haversine distance has been chosen to calculate the distance between the nodes. Haversine distance chosen to provide the accurate measure distance two point on the earth surface, considering curvature of the earth. Each ant's movement in this algorithm will be used in the Haversine distance to get the total distance in kilometres.

2.3 Related Data

The necessary data and information for this study were obtained from the logistics department of a Food Industry in Guar Chempedak, Kedah. The collected data includes the locations of retail outlets, specific customer demands, and their geographical coordinates, represented by longitude and latitude. The Ant Colony Optimization (ACO) algorithm was employed as the primary technique and implemented using Python software. The company sources its raw materials from three primary suppliers: one provides flour, the other supplies spices, and the third delivers oil, all of which is crucial for food production. Once the manufacturing process is completed, the food products are distributed from the company to 19 retail outlets. Fig. 1 below illustrates the flow configuration of the supply chain process, providing a general overview rather than the actual detailed process.

Enhanced Knowledge in Sciences and Technology Vol. 5 No. 2 (2025) p. 101-111

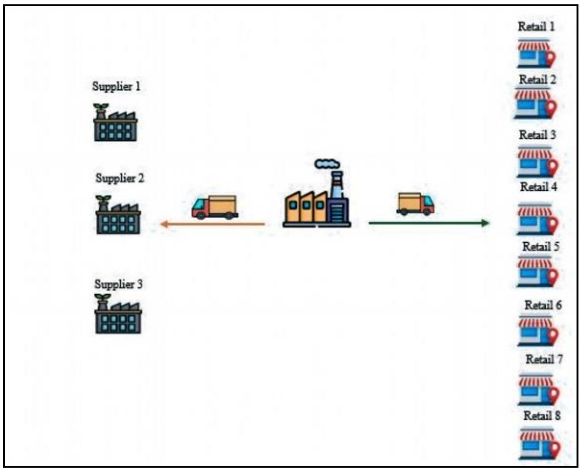

Fig. 1 Configuration of the proposed model

This study focuses on the delivery routes within the company to retail locations in Kedah. The distribution process operates from a single warehouse, with each trip covering 3 to 5 retail nodes. Specifically, Trip 1 on Day 1 covers deliveries to 3 retail locations, Trip 2 on Day 2 covers 4 locations, Trip 3 on Day 3 covers 3 locations, Trip 4 on Day 4 covers 4 locations, and Trip 5 accommodates deliveries to 5 retail locations. A detailed summary of the daily trips is presented in Table 1.

Table 1 *Daily trip summary*

Day	Number of Trips	Number of The Retails
1	Trip 1	3
2	Trip 2	4
3	Trip 3	3
4	Trip 4	4
5	Trip 5	5

The warehouse of the company is in the Guar Chempedak region with the postcode 08800, which is strictly in Kedah. The selected customer locations include Kulim, Kodiang, Jitra, Kampung Bukit Selambau, Pokok Sena, Sik, Amanjaya Sungai Petani, Bertam, Taman Setia Pendang, Taman Putri Lagenda Padang Serai, Tanah Merah Pendang, Kuala Nerang Padang Terap, Alor Setar, Jitra, Gurun, Baling, Sungai Petani, and Pendang. Notably, Alor Setar has been repeated as it includes multiple delivery points in different areas. All locations are in Kedah. Deliveries for the retailers will follow the trip that has been organized. Fig. 2 shows the locations and regions of the delivery.

Fig. 2 *Location of the customers*

The latitude and longitude coordinates for each node were obtained using the Global Positioning System (GPS). Coordinate Malaysia. These nodes play a critical role in identifying the optimal delivery routes. Table 2 shows all the retail locations and regions of the delivery.

Table 2 Latitude and Longitude for all retail

Retail	Location	Latitude	Longitude	Demand
1	Kulim	5.365	100.561	300
2	Kodiang Jitra	6.3198	100.4241	200
3	Kampung Bukit Selambau	5.7429	100.5554	200
4	Pokok Sena	6.2086	100.6113	500
5	Sik	5.8093	100.7737	600
6	Alor Setar	6.1214	100.3674	300
7	Amanjaya Sungai Petani	5.6491	100.4913	500
8	Bertam	5.4815	100.5024	700
9	Taman Setia, Pendang	5.9891	100.4764	400
10	Taman Putri Lagenda, Padang Serai	5.4826	100.5213	100
11	Tanah Merah, Pendang	5.9831	100.4901	200
12	Kuala Nerang, Padang Terap	6.2637	100.5784	150
13	Alor Setar	6.1214	100.3674	200
14	Jitra	6.267	100.431	300
15	Gurun	5.8176	100.4704	200
16	Baling	5.6759	100.9185	200
17	Sungai Petani	5.647	100.4876	500
18	Alor Setar	6.1214	100.3674	500
19	Pendang	5.9891	100.4764	300

The collected data will be integrated into a Python-based algorithm to identify the most suitable and efficient delivery solution. Table 3 outlines the total delivery distances for all five scheduled trips, calculated without the application of any optimization algorithm. These initial results serve as benchmarks, with their main attributes summarized in the table. To assess the improvements achieved by implementing the ACO algorithm, the computational results will be analysed using the Relative Percentage Deviation (RPD) method.

The (RPD) is calculated using a specific formula to compare the differences between the ACO algorithm and the Multiple Ant Colony algorithm.

$$RPD = \frac{Benchmark\ Distance - ACO\ Distance}{Benchmark\ Distance} \times 100\%$$

Table 3 Characteristics of Benchmark Solutions

Trip	Number of Retail	Demand	Distance KM
1	3	700	353.8
2	4	1900	277
3	3	1200	115.45
4	4	850	123.0
5	5	1700	224.7

3. Results and Discussion

3.1 Optimal Solution

The algorithm was running 6 times for each trip or instance to obtain the greatest result and by setting the parameter with ants' number = 10, maximum iteration = 200, α = 1, β=2 and $\boldsymbol{q_0}$= 0.1. The distance for every path or node is calculated using the haversine formula in km. Table 4 shows the summary of the results for the vehicle routes calculated using Python, while Tables 5 to 9 show the original details of each route plan, which will be compared after optimization. All deliveries must be made between 9:00 AM and 5:00 PM. Tables 3 through 7 show the details for each trip and are used for comparison after the optimization process.

Table 4 Summary of the result optimal solution

Trip	Total Distance (km)	Relative Percentage Deviation (RPD)%	Result (Routes)
1	215.25	39.19 %	Warehouse → R2 → R3 → R1 → Warehouse
2	167.01	39.75 %	Warehouse → R7 → R5 → R4 → R6 → Warehouse
3	115.45	29.04 %	Warehouse → R9 → R10 → R8 → Warehouse
4	113.04	8.09 %.	Warehouse → R11 → R12 → R14 → R13 → Warehouse
5	181.56	19.22 %	Warehouse → R15 → R17 → R16 → R19 → R18 → Warehouse

Table 5 Detail for each node and location Trip 1

Retail	Demand	Time Delivery	Service Time
Warehouse	0	9 AM - 5 PM	0
1	300	9 AM - 5 PM	1 hour 6 minutes
2	200	9 AM - 5 PM	1 hour 42 minutes
3	200	9 AM - 5 PM	42 minutes

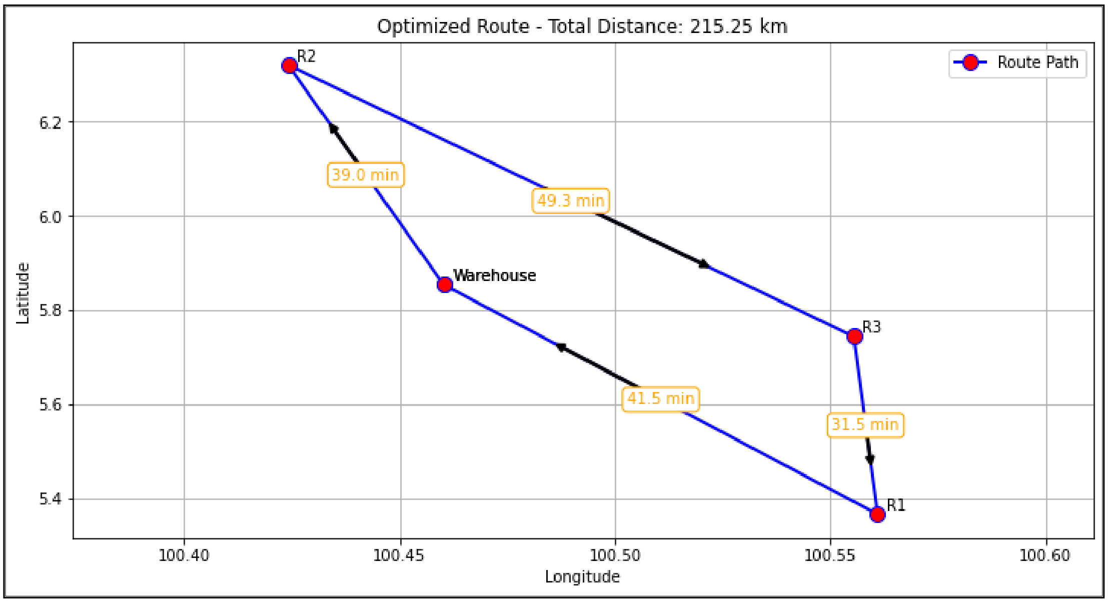

Fig. 3 The vehicle route plan for Trip 1

Fig. 3 shows relate the explanation with Table 5 that the plan in Trip 1 was optimized to meet time constraints and vehicle capacity, with Python used to calculate the results. The vehicle covered a total distance of 215.25 km. The journey started at the warehouse, delivering goods to R2, followed by R3 and R1, before returning to the warehouse. The entire trip took 6.19 hours, with the following breakdown: from the warehouse to R2, a distance of 52.06 km was covered in 39.04 minutes, arriving at 09:39; from R2 to R3, the vehicle travelled 65.77 km in 49.33 minutes, reaching R3 by 12:10; the trip from R3 to R1 was 42.03 km, taking 31.52 minutes, with arrival at 13:23; and from R1 back to the warehouse, the vehicle covered 55.40 km in 41.55 minutes, arriving at 15:11. Additionally, the Relative Percentage Deviation calculation showed a 39.19% reduction in the total distance. The optimization successfully improved the route's efficiency, reducing both time and distance while ensuring timely delivery.

Table 6 Detail for each node and location Trip 2

Retail	Demand	Time Delivery	Service Time
Warehouse	0	9 AM - 5 PM	0
4	500	9 AM - 5 PM	54 minutes
5	600	9 AM - 5 PM	30 minutes
6	300	9 AM - 5 PM	1 hour 12 minutes
7	500	9 AM - 5 PM	48 minutes

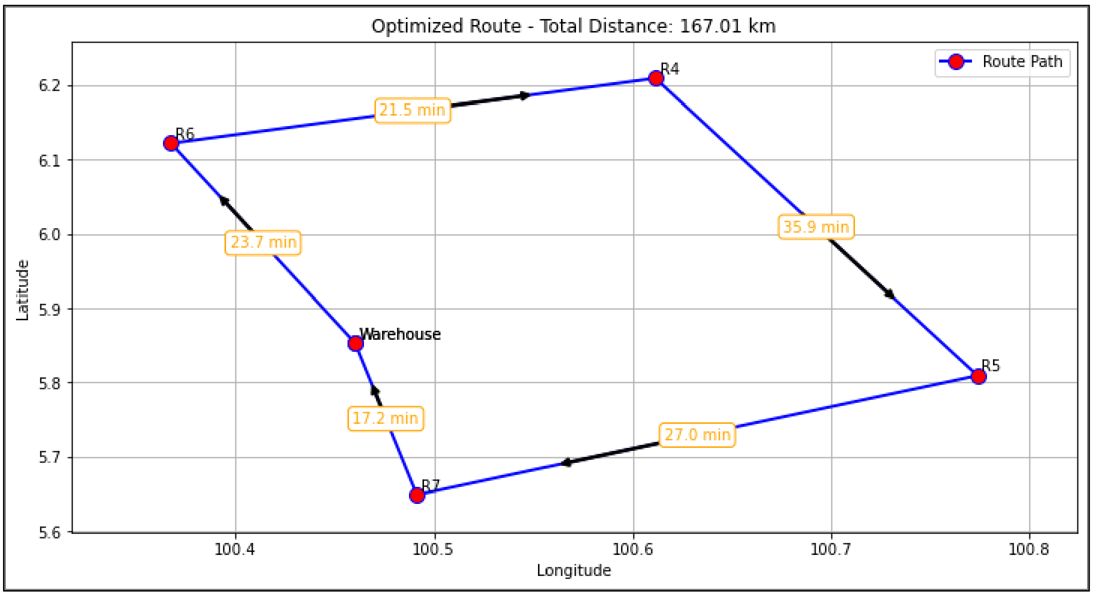

Fig. 4 The vehicle route plan for Trip 2

The plan relates the explanation with Table 6 and Fig. 4 in Trip 2 was optimized using Python to account for time constraints and vehicle capacity, covering a total of 167.01 km. The route started at the warehouse, delivering goods to R6, then to R4, R5, R7, and back to the warehouse. The journey took 5.49 hours, with each segment as follows: warehouse to R6 (31.56 km, 23.67 minutes, arrival at 09:17), R6 to R4 (28.65 km, 21.49 minutes, arrival at 10:32), R4 to R5 (47.89 km, 35.92 minutes, arrival at 11:38), R5 to R7 (35.97 km, 26.97 minutes, arrival at 12:53), and R7 to the warehouse (22.93 km, 17.20 minutes, arrival at 14:29). The optimization resulted in a 39.75% reduction in total distance, improving both time and delivery efficiency.

Table 7 Detail for each node and location Trip 3

Retail	Demand	Time Delivery	Service Time
Warehouse	0	9 AM - 5 PM	0
8	700	9 AM - 5 PM	1 hour
9	400	9 AM - 5 PM	36 minutes
10	100	9 AM - 5 PM	1 hour 18 minutes

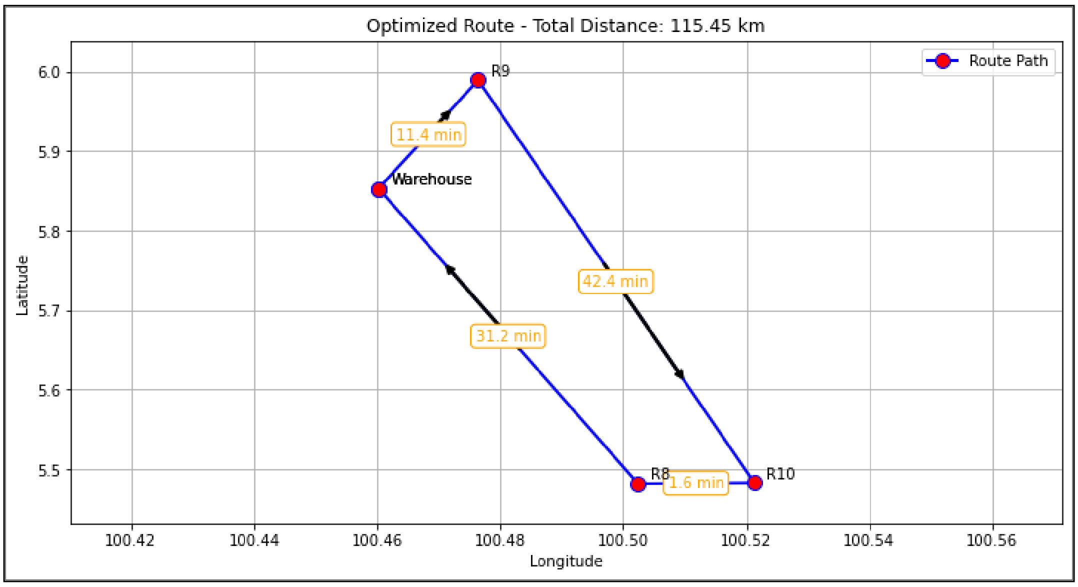

Fig. 5 The vehicle route plan for Trip 3

Fig. 5 relate the explanation with Table 7 illustrates the route for the vehicle in Trip 3 was optimized using Python, considering time constraints and vehicle capacity, covering 115.45 km. The trip began with a delivery from the warehouse to R9, followed by R10, R8, and then back to the warehouse. The total trip duration was 4.34 hours, with each segment as follows: warehouse to R9 (15.24 km, 11.40 minutes, arrival at 09:11), R9 to R10 (56.54 km, 42.40 minutes, arrival at 10:29), R10 to R8 (2.10 km, 1.6 minutes, arrival at 11:49), and R8 to the warehouse (41.57 km, 31.2 minutes, arrival at 13:20). The optimization led to a 29.04% reduction in total distance.

Table 8 Detail for each node and location Trip 4

Retail	Demand	Time Delivery	Service Time
Warehouse	0	9 AM - 5 PM	0
11	200	9 AM - 5 PM	42 minutes
12	150	9 AM - 5 PM	1 hour 30 minutes
13	200	9 AM - 5 PM	1 hour 12 minutes
14	300	9 AM - 5 PM	1 hour 24 minutes

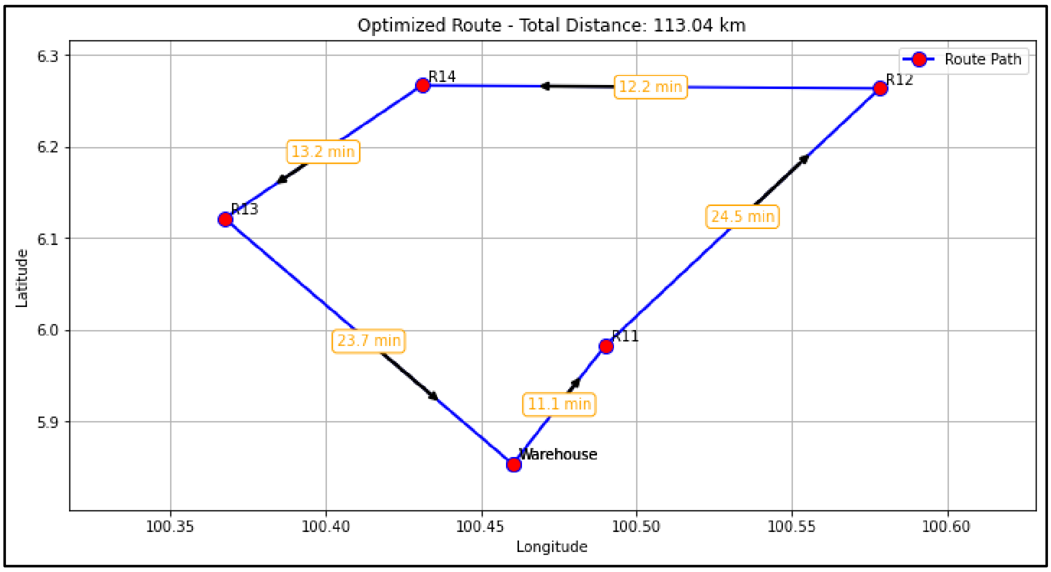

Fig. 6 *The vehicle route plan for Trip 4*

Penerbit UTHM

Fig. 6 relates the explanation with Table 8 covering a total of 113.04 km. The journey began with a delivery from the warehouse to R11, followed by R12, R14, R13, and finally returning to the warehouse. The total trip duration was 6.21 hours, with each segment as follows: warehouse to R11 (14.84 km, 11.13 minutes, arrival at 09:11), R11 to R12 (32.69 km, 24.52 minutes, arrival at 10:17), R12 to R14 (16.30 km, 12.22 minutes, arrival at 11:59), R14 to R13 (17.65 km, 13.24 minutes, arrival at 13:37), and R13 to the warehouse (31.56 km, 23.67 minutes, arrival at 15:12). The optimization resulted in an 8.09% reduction in total distance, demonstrating improved efficiency.

Table 9 Detail for each node and location Trip 5

Retail	Demand	Time Delivery	Service Time
Warehouse	0	9 AM - 5 PM	0
15	200	9 AM - 5 PM	24 minutes
16	200	9 AM - 5 PM	1 hour 6 minutes
17	500	9 AM - 5 PM	54 minutes
18	500	9 AM - 5 PM	1 hour 12 minutes
19	300	9 AM - 5 PM	30 minutes

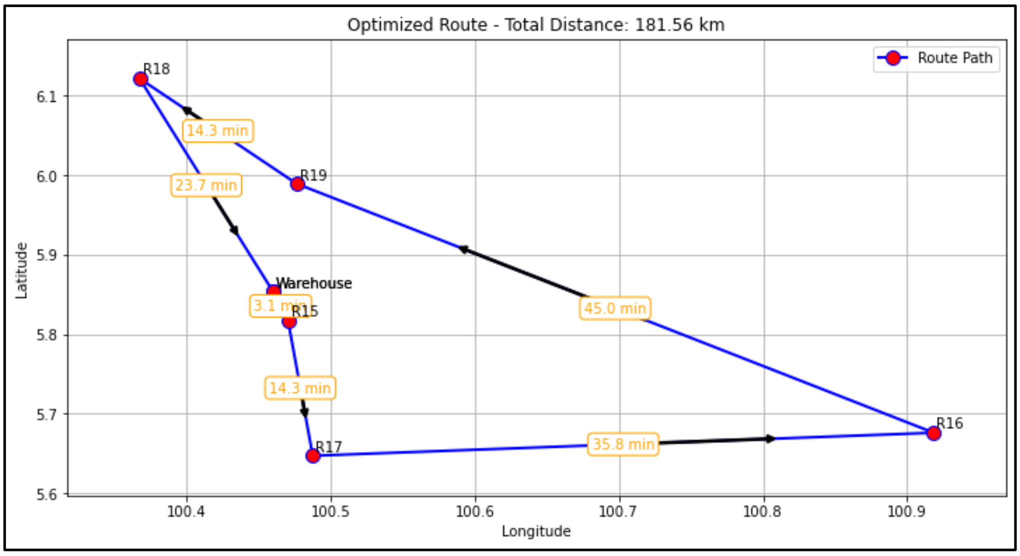

Fig. 7 The vehicle route plan for Trip 5

The route relates the explanation with Table 9 and Fig. 7 for the vehicle in Trip 5 was optimized using Python, factoring in time limitations and vehicle capacity, covering a total of 181.56 km. The journey started with a delivery from the warehouse to R15, followed by R17, R16, R19, R18, and a return to the warehouse. The total trip duration was 6.37 hours, with the following segment details: warehouse to R15 (4.09 km, 3.07 minutes, arrival at 09:03), R15 to R17 (19.07 km, 14.30 minutes, arrival at 09:41), R17 to R16 (47.79 km, 35.84 minutes, arrival at 11:11), R16 to R19 (60.04 km, 45.03 minutes, arrival at 13:02), R19 to R18 (19.02 km, 14.26 minutes, arrival at 13:46), and R18 to the warehouse (31.56 km, 23.67 minutes, arrival at 15:22). The optimization resulted in a 19.22% reduction in total distance, reflecting the improved efficiency.

4. Conclusions and Recommendations

This study successfully achieved its objectives of optimizing vehicle routing and scheduling in supply chain management using the Ant Colony Algorithm. The dataset, obtained from the food industry in Kedah, included 19 retail customer locations with demands and time windows. Python software was used to run the algorithm, resulting in optimal delivery distances which are 215.25 km in Trip 1, 167.01 km in Trip 2, 115.45 km in Trip 3, 113.04 km for Trip 4, and 181.56 km for Trip 5. The results have been verified and validated by comparing the results obtained using the Ant Colony Optimization (ACO) algorithm with benchmark solutions. A benchmark solution is a standard reference used to compare and verify the accuracy, efficiency, and reliability of the ACO algorithm, typically derived from exact methods, existing heuristics, or real-world data. These findings demonstrate the effectiveness of the Ant Colony Algorithm in minimizing travel distances.

Enhanced Knowledge in Sciences and Technology Vol. 5 No. 2 (2025) p. 101-111

As a recommendation, incorporating realistic road network data and establishing strategic delivery hubs closer to customers are suggested to improve route accuracy, reduce delivery times, and enhance logistics efficiency.

Acknowledgement

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

Conflict of Interest

The 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:*** *Nur Aqilah Nadirah Che Amran, Siti Suhana Jamaian;* ***data collection****: Nur Aqilah Nadirah Che Amran;* ***analysis and interpretation of results:*** *Nur Aqilah Nadirah Che Amran, Siti Suhana Jamaian;* ***draft manuscript preparation:*** *Nur Aqilah Nadirah, Siti Suhana Jamaian. All authors reviewed the results and approved the final version of the manuscript*

References

[1] M. C. Cooper, D. M. Lambert, and J. D. Pagh, “Supply Chain Management: More Than a New Name for Logistics,” *The International Journal of Logistics Management*, vol. 8, no. 1, pp. 1–14, Jan. 1997, Doi: 10.1108/09574099710805556.

[2] C. A. Silva, J. M. C. Sousa, T. A. Runkler, and J. M. G. Sá da Costa, “Distributed Supply Chain Management Using Ant Colony Optimization,” *Eur J Oper Res*, vol. 199, no. 2, pp. 349–358, Dec. 2009, Doi: 10.1016/j.ejor.2008.11.021.

[3] P., & V. D. (Eds.). Toth, “Vehicle Routing Problems, Methods, and Applications,” *Society for Industrial and Applied Mathematics*, vol. 2, no. 1, pp. 463–465, 2014, Doi: 103.130.13.130.

[4] M. Dorigo, “Ant Colony Optimization,” *Scholarpedia*, vol. 2, no. 3, pp. 14–61, 2007, Doi: 10.4249/scholarpedia.1461.

[5] L. Bianchi *et al.*, “Hybrid Metaheuristics for The Vehicle Routing Problem with Stochastic Demands,” *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 1, pp. 91–110, Apr. 2006, Doi: 10.1007/s10852-005-9033-y.

[6] M. Gendreau, F. Guertin, J. Y. Potvin, and R. Séguin, “Neighborhood Search Heuristics for A Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries,” *Transp Res Part C Emerg Technol*, vol. 14, no. 3, pp. 157–174, 2006, Doi: 10.1016/j.trc.2006.03.002.

[7] Md. R. I. Islam, Md. E. I. Monjur, and T. Akon, “Supply Chain Management and Logistics: How Important Interconnection Is for Business Success,” *Open Journal of Business and Management*, vol. 11, no. 05, pp. 2505–2524, 2023, Doi: 10.4236/ojbm.2023.115139.

[8] A. Delgoshaei, M. K. M. Ariffin, B. T. H. T. Bin Baharudin, and Z. Leman, “Minimizing Makespan of a Resource-Constrained Scheduling Problem: A Hybrid Greedy and Genetic Algorithms,” *International Journal of Industrial Engineering Computations*, vol. 6, no. 4, pp. 503–520, 2015, Doi: 10.5267/j.ijiec.2015.5.002.

[9] M., M. & W. Ibrahim, “An Improved Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows,” *Jurnal Teknik Industri*, vol. 23, no. 2, pp. 105–120, 2022, Doi: 10.22219/JTIUMM.Vol23.No2.105-120.

[10] T., & P. V. Carwalo, “Solving Vehicle Routing Problem using Ant Colony Optimization with Nodal Demand,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 9, pp. 181–278, 2015, [Online]. Available: www.ijert.org