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

Vol. 4 No. 2 (2024) 45-55
https://publisher.uthm.edu.my/periodicals/index.php/ekst

EKST

A Capacitated Vehicle Routing Problem with Time Windows for Fleet Management Using Ant Colony Algorithm

Muhammad Afiq[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.2024.04.02.006

Article Info

Received: 27 December 2023
Accepted: 11 January 2024
Available online: 12 December 2024

Keywords

Capacitated Vehicle Routing Problem, Time Windows , Ant Colony algorithm, Fleet Management.

Abstract

Fleet management has an important role in the supply chain for enhancing business activity. Fleet management can face big problems without effective planning in scheduling and choosing the best routes for delivery. The choice of the best distance path is crucial in ensuring efficient deliveries. In the raised problem, our objective is to minimize the distance, the fuel cost, and the number of vehicles used. Since the vehicle routing problem is Nondeterministic Polynomial (NP)-hard in nature of the problem, the ant colony optimization is used to determine the optimal solution with the assistance of Python software. This study applies an ant colony algorithm to solve a capacitated vehicle routing problem with time windows for fleet management. The warehouse of the manufacturing industry is at Kempas Lama, while the location of the customers included under consideration in our study located at Gong Badak, Ipoh, Semenyih, Mentakab, Kota Bharu, Muar, Ayer Hitam, Seremban, Kuala Selangor, Kuantan, Bukit Bendera, Segamat and Tanjung Minyak. There are five planned trips for delivery, with one planned trip for one day. The variable vehicle capacity, time windows, and also the customer's demand were considered to obtain the optimal solution. The results of the optimal shortest distance for delivery are 1210.92 km, 1776.65 km, 1428.54 km, 2204.81 km, and 707.45 km for the five trips, respectively. Some of the variables have been adjusted in the solution to show the more efficient short distance. This research provides substantial insights into enhancing fleet management efficiency through algorithmic optimization.

1. Introduction

Fleet management is defined as satisfying changing client needs in short service windows, which calls for cost-effectiveness, asset and driver management, operational efficiency, and compliance [1]. Fleet management is a rapidly evolving field that is expected to expand at a compound annual growth rate (CAGR) of 15.5% to reach USD 52.4 billion by 2027 [2]. As diesel prices continue to remain high this year, improving fuel efficiency and reducing fuel costs have become top priorities for fleet management. Therefore, any increase in fuel prices can have a profound impact on profitability and overhead costs for carriers and smaller shipping businesses.

The fleet management industry continues to grow steadily, striving to enhance productivity and find effective solutions for industry challenges. To achieve this, the ant colony algorithm has been selected as a monitoring, rescheduling, and optimization technique for fleet management. The algorithm aims to minimize costs and total travel time, enabling efficient distribution and collection of products from one or more depots to clients [3]. The capacitated vehicle routing problem (VRP) is a VRP in which vehicles with a constrained carrying capacity must pick up or deliver things at multiple places. Capacitated VRP is strongly Nondeterministic Polynomial (NP)-hard because it includes the traveling salesman problem as a special case. However, tiny to medium-sized cases can be resolved to with known optimality [4].

One of the critical problems is the VRP, which is to find the best distribution routes from depots to various clients while taking constraints into account [5]. The goal of the VRP is to identify K identical vehicles based at the depot a set of routes that minimizes the overall routing cost while visiting each vertex precisely once. Numerous variations of this traditional composition have also been researched. The capacitated VRP, where each customer has a demand for a good, and vehicles have finite capacity; the VRP with Time Windows, where each customer must be visited during a certain time period; the VRP with Pick-up and Delivery, where goods must be picked up and delivered in specific amounts at the vertices; and the Heterogeneous fleet VRP, where vehicles have varying capacities, are among the most popular [6]. The VRP with time windows seeks an optimal set of tours that a fleet of vehicles should do in order to service a group of consumers within the designated time windows. According to a recent review of the literature, it represents a significant variation of the well-known VRP, one of the most active areas in operations research [7].

In this study, a capacitated VRP with time windows is applied to address fleet management challenges experienced by the manufacturing industry. The manufacturing industry is located at Kempas Lama, Johor, and the study will be conducted within the area encompassed by the manufacturing industry fleet delivery services. For the purpose of this study, the manufacturing industry is chosen because of its possession of a private fleet vehicle and involvement in delivery services for distributing its products to customers. The main objective of this study is to minimize the total cost for all routes while taking into consideration capacity restrictions and time windows for service delivery. To achieve this goal, Python is utilized in conjunction with the ant colony algorithm, as an effective approach for solving the capacitated VRP with time windows. The combination of Python and the ant colony algorithm enables the optimization of route planning and resource allocation, resulting in efficient and cost-effective fleet management for the selected manufacturer industry.

2. Methodology

This study focuses on describing the mathematical model utilized to address the challenges of a capacitated Vehicle Routing Problem (VRP) with time windows for fleet management applications. Since the VRP falls under the category of nondeterministic polynomial (NP)-hard problems, the ant colony algorithm is chosen as the solution approach. It is crucial to incorporate the Ant Colony algorithm in order to achieve efficient fleet management and optimize transportation operations for the industry. Python is selected as the solver to tackle this problem, leveraging its capabilities for efficient algorithm implementation and optimization. The research is conducted within a production industry located in Johor, with the aim of gathering relevant data specific to the industry's transportation needs.

2.1 Capacitated vehicle routing problems with time windows

In this study, the capacitated vehicle routing problem with time windows for fleet management holds significant importance in real-world applications. The primary significance and goal of this study lie in determining optimal routes and scheduling fleet vehicles while considering capacity and time restrictions. By focusing on these aspects, this study aims to meet customer satisfaction by ensuring deliveries are made within specified time windows since the time frame constraint is crucial as it directly impacts customer expectations.

Moreover, this study holds significance in reducing costs, particularly fuel expenses. Thus, optimized routes have the potential to yield substantial cost savings in fleet management. To achieve cost savings, this study aims to develop algorithms and models that optimize routes and minimize fuel consumption. By utilizing the ant colony algorithm, this study strives to create a system that effectively balances route lengths, capacity constraints, and time windows, leading to optimized fleet operations.

2.2 Objective function

The capacitated VRP with times windows can be represented as a weighted directed graph that represents the whole set of vertices and represents the set of arcs between the vertices. The depot is assigned as 0 and the others j represent the customer while I represent the current node. Each client has a 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 mathematically be represented as [8]:

$$\text{Minimize } Z = \sum_{i=0}^{NN} \sum_{j=0}^{NN} \sum_{k=1}^{NV} y_i^k x_{ij}^k \tag{1}$$

Enhanced Knowledge in Sciences and Technology Vol. 4 No. 2 (2024) p. 45-55

Which is, it refers to minimizing the total distance, where *NN* refers to the number of nodes, *NV* refers to the number of vehicles, x_{ij}^{k} refers to the vehicle *k* visits the customers j directly after the customers i and y_i^k refers to the customers i is served by the vehicle k (cost of travel from node *i* to node *j*).

2.2.1 Constraints

The constraints considered in this study are as follows:
For some of the variables that have in the equation can be referred to as *NN* is the numbers of nodes, *NV* is the number of the vehicle, x_{ij}^{k} is the vehicle *k* visits the customers j directly after the customers, i , and y_i^k is the customers i is served by the vehicle k is 1 (cost of travel from node i to node j).

Constraint (2) is meant to force the number of trucks flowing into and out of node *j* need to be the same.

$$\sum_{i=0}^{NN}\sum_{k=1}^{NV} x_{ij}^{k} = \sum_{i=0}^{NN}\sum_{k=1}^{NV} x_{ij}^{k} \quad \text{for all } j \tag{2}$$

Constraint (3) states that the demand at every node can be satisfied by one vehicle *k* or a single route needed to give to each customer [8].

$$\sum_{i=0}^{NN}\sum_{k=1}^{NV} x_{ij}^{k} = 1 \quad \text{for all } j \tag{3}$$

Constraint (4) represents that each route is covered by one vehicle.

$$\sum_{i=0}^{NN}\sum_{k=0}^{NN} x_{ij}^{k} \leq 1 \quad \text{for all } k \tag{4}$$

Constraint (5) describes sub-tour elimination constraint and capacity constraint.

$$u_{ik} - u_{jk} + Qx_{ij}^{k} \leq Q - q_i \text{ for all } i, j \in N(k), k \tag{5}$$

where $u_{ik} - u_{jk}$ is the parameter that show the load cumulative after delivery *Q* is capacity and q_i is demand at every node. *N(k), k* is the node that is associated with vehicle *k.* It means the possibility of its pair node within *N(k).*
Constraint (6) shows that the vehicle *k* is not allowed to visit all the nodes more than its capacity.

$$Q - \sum_{j=1}^{NN}\left\{q_i \sum_{j=0}^{NN} x_{ij}^{k}\right\} \geq 0 \quad \text{for all } k \tag{6}$$

where *Q* is capacity of the vehicle and q_i is the demand at every node.
Constraints (7) and (8) specify that the vehicle *k* should start and end at the depot [9].

$$\sum_{i=1}^{NN} x_{ij}^{k} - y_i^k = 0 \quad \text{for all } k \tag{7}$$

$$\sum_{j=1}^{NN} x_{ij}^{k} - y_i^k = 0 \quad \text{for all } k \tag{8}$$

2.3 Ant colony algorithm

The ant colony optimization (ACO) algorithm is used to determine and optimize the total of all routes while considering capacity and time limits as restrictions. In this method, the vehicle routing problem (VRP) with the variant capacitated VRP with time windows was studied. The ACO algorithm is applied to solve the capacitated VRP with time windows. The algorithm will minimize the routes and if the result does not show the best outcome, the ACO algorithm will be applied again. To solve the capacitated VRP efficiently for fleet management, the ACO algorithm is utilized with the assistance of Python. The ACO algorithm can be implemented as follows [10]:

Step 1: Initialize all the parameters that will be used.
Step 2: The ant is located at the starting point.
Step 3: Ant *k* will apply a probabilistic called a random proportional rule to choose or decide the next destination or node. For each ant *k* the probability $p_{k(i,j)}$ of moving from the current node i to another node or destination is calculated taking to the formula where β and α are parameters. τ_{ij} is the amount of the pheromone between the nodes *i* and *j:*

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{j \in N_i^k} \tau_{ij}^{\alpha}\eta_{ij}^{\beta}} \tag{9}$$

where η_{ij} is for the heuristic and the value of η_{ij} is calculated as follows:

$$\eta_{ij} = \left\{ \left[\left(\max\left\{ t_{ij}(t_i) \right\}, e_j - t_i \right)^{a} . (l_j - t_i)^{b} \right]^{1/(a+b)} \right\}^{-1} \tag{10}$$

where $t_{ij}(t_i)$ is the arrival time to j at time t_i.

Step 4: If the vehicle meets with the next client before the time windows, $e_j - t_i$ will be chosen and $l_j - t_i$ is time between the departure and the latest time. e_j and l_j refers earliest and the latest time.

Step 5: Ant k selects the customers j by following the pseudorandom proportional rule.

$$j = \begin{cases} \arg\max_{l \in N_i^k} \left\{ \tau_{il}\eta_{il}^{\beta} \right\} & if \;\; q \le q_0 \\ J & if \;\; q \ge q_0 \end{cases} \tag{11}$$

Step 6: When ant k constructs the route, the pheromone is updated with the following formula:

$$\tau_{ij} = (1-\varphi)\tau_{ij} + \varphi\tau \qquad (0 \le \varphi \le 1) \tag{12}$$

where the value of τ_0 is the same as the initial value for the pheromone trails and φ is pheromone evaporation.

Step 7: When the ants k colony have completed their computation, the best tour will be selected from all the iteration and globally modify the pheromone with the following formula:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{best} \qquad (0 \le \rho \le 1) \tag{13}$$

to modify the pheromone trail where $\rho\Delta\tau_{ij}^{best} = 1/L_{best}$. L_{best} is the length of iteration best tour and ρ is the pheromone evaporation rate.

For this algorithm, Haversine distance has been chosen to calculate the distance between the nodes. Haversine distance was chosen to provide the accurate measure of distance between two points on the earth's surface, considering the curvature of the earth. Every ant's movement in this algorithm will use the haversine distance to get the total distance in kilometres.

2.4 Data for fleet routing and schedule

This research focused on the delivery trip of soy sauce manufacturing industry products to customers involved in Peninsular Malaysia. There are 5 Trips planned for each day, which to specify that for Trip 1 day 1 has 3 locations of customers, Trip 2 for day 2 has 4 locations of customers, Trip 3 for day 3 has 5 locations of customers, Trip 4 for day 5 has 5 locations of customers and Trip 5 for day 5 has 3 locations of customers. The summary of the daily trip as in Table 1 and the location of the customer can be seen in Figure 1.

Table 1 Daily trip summary

Day	Trip Number	Number of Customers
1	Trip 1	3
2	Trip 2	4
3	Trip 3	5
4	Trip 4	5
5	Trip 5	3

The warehouse of the manufacturing industry is located at Kempas Lama, Johor while the location of the customers includes Gong Badak, Ipoh, Semenyih, Mentakab, Kota Bharu, Muar, Ayer Hitam, Seremban, Kuala Selangor, Kuantan, Bukit Mertajam, Segamat and Tanjung Minyak. All locations are located in peninsular Malaysia. The delivery for the customers will follow the trip that has been organized. Fig. 1 shows the state customers involve and regions of the delivery.

Fig. 1 Location of the customers

Data latitude and longitude for each node are obtained from the manufacturing industry. Each node is needed to find the optimum solution for the delivery. Table 2, Table 3, Table 4, Table 5, and Table 6 show the coordinates for each node for the customers to deliver the product and the details about the customers. The details include the service time, demand, and time windows. Service time for all trips is 90 minutes, and time windows for all trips are 0 until 1440 minutes, referred to 24 hours. Demand for each trip also can be seen in Table 2, Table 3, Table 4, Table 5, and Table 6 for Trip 1, Trip 2, Trip 3, Trip 4, and Trip 5, respectively. All this information will be used in the algorithm with Python to obtain the best short distance for optimal solution.

Table 2 Latitude longitude for each node in Trip 1

Node	Location	Latitude	Longitude	Demand	Ready Time	Due Time	Service Time
0	Warehouse	1.556700706	103.7153931	0	0	1440	90
1	Kota Bharu	5.402130200	102.0635972	1200	0	1440	90
2	Muar	2.050000000	102.5666667	600	0	1440	90
3	Ayer Hitam	1.918172800	103.1795162	600	0	1440	90

Table 3 Latitude longitude for each node in Trip 2

Nodes	Location	Latitude	Longitude	Demand	Ready Time	Due Time	Service Time
0	Warehouse	1.556700706	103.7153931	0	0	1440	90
1	Gong Badak	5.385760000	102.9949297	1200	0	1440	90
2	Ipoh	4.598681700	101.0900236	400	0	1440	90
3	Semenyih	2.947391100	101.8459911	400	0	1440	90
4	Mentakab	3.486451000	102.3514872	400	0	1440	90

Table 4 Latitude longitude for each node in Trip 3

Nodes	Location	Latitude	Longitude	Demand	Ready Time	Due Time	Service Time
0	Warehouse	1.556700706	103.7153931	0	0	1440	90
1	Semenyih	2.9473911	101.8459911	300	0	1440	90
2	Seremban	2.7184070	101.9410696	300	0	1440	90
3	K. Selangor	3.3621022	101.3455503	300	0	1440	90

4	Ipoh	4.5986817	101.0900236	300	0	1440	90
5	Kuantan	3.7985637	103.3219900	1200	0	1440	90

Table 5 Latitude longitude for each node in Trip 4

Nodes	Location	Latitude	Longitude	Demand	Ready Time	Due Time	Service Time
0	Warehouse	1.556700706	103.7153931	0	0	1440	90
1	Ipoh	4.598681700	101.0900240	300	0	1440	90
2	Gong Badak	4.863074300	102.9949300	300	0	1440	90
3	B. Bendera	5.406501300	100.2559080	1200	0	1440	90
4	Semenyih	2.947391100	101.8459910	300	0	1440	90
5	Segamat	2.500491400	102.8156550	300	0	1440	90

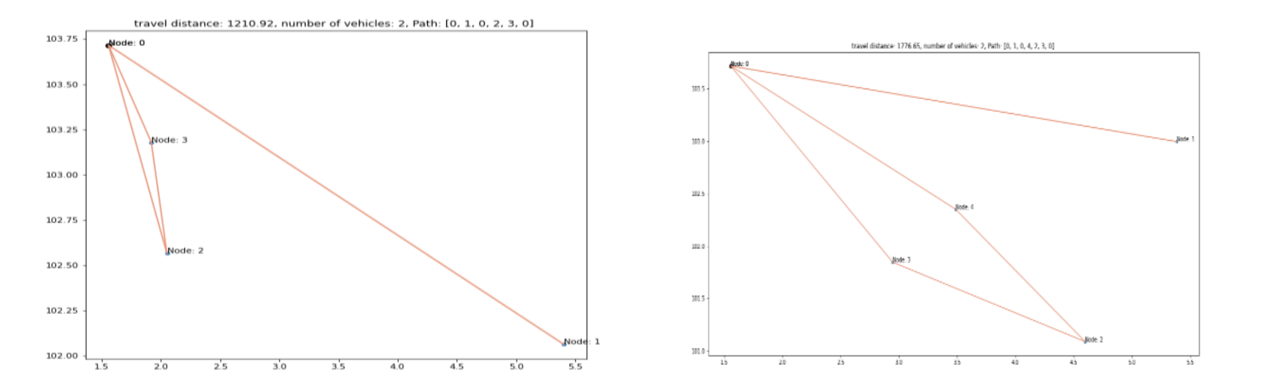
Table 6 Latitude longitude for each node in Trip 5

Nodes	Location	Latitude	Longitude	Demand	Ready Time	Due Time	Service Time
0	Warehouse	1.556700706	103.7153931	0	0	1440	90
1	Kuantan	3.798563700	103.3219900	400	0	1440	90
2	Semenyih	2.947391100	101.8459911	400	0	1440	90
3	Tg Minyak	2.329374400	102.2880962	400	0	1440	90

For information in Table1, Table2, Table3, Table 4 and Table 5, certain locations need to deliver multiple times per week due to customer preferences for specific days and demand because of their limited capacity of the store.

3. Result and Discussion

The algorithm was running 10 times for each trip to obtain the greatest result and by setting the parameter with ants number = 10, maximum iteration = 200, α = 1, β=2, and q_0 = 0.1 that will be applied in (9) until (13). The distance for every path or nodes is calculated using the haversine formula in km. The result of the optimal solution can be shown in Fig. 2.



(a) (b)

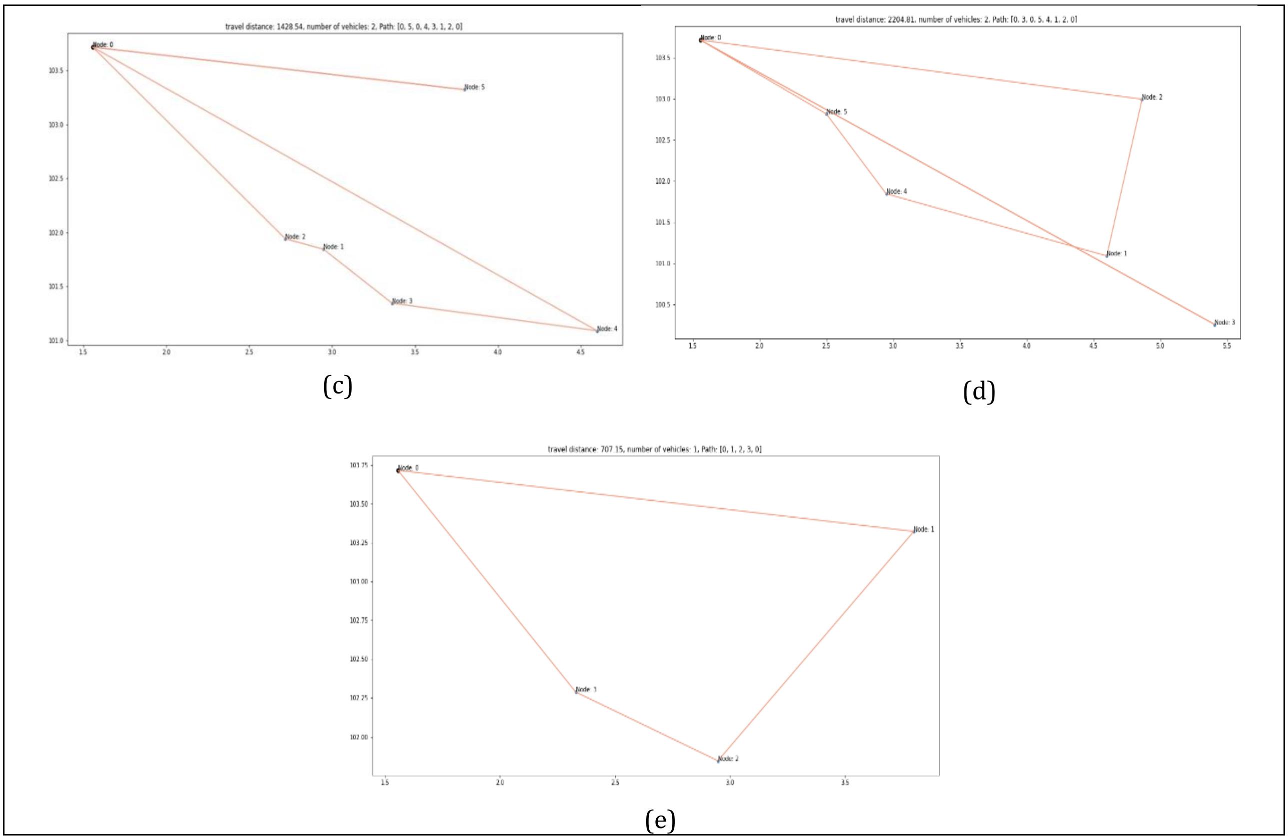


Fig. 2 The planned trips for (a) Trip 1, (b) Trip 2, (c) Trip 3, (d) Trip 4 and (e) Trip 5

Fig. 2 shows the best distance for Trip 1, Trip 2, Trip 3, Trip 4, and Trip 5. Trip 1 is 1210.92 km, with the total number of vehicles used for the delivery to the customers being 2. Trip 2 is 1776.65 km, with the total vehicles used for the delivery to the customers is 2. Trip 3 is 1428.54 km, with the total vehicle used for the delivery to the customers is 2. Trip 4 is 2204.41km, with the total vehicle be used for the delivery to the customers is 2. Trip 5 is 707.45 km with the total vehicles used for the delivery to the customers is 2.

The optimal solution for Trip 1, Trip 2, Trip 3, Trip 4 and Trip 5 is summarized in Table 7. The trip is used 2 vehicles except Trip 5 due to the location of the customers in Trip 5 can handle only 1 vehicle to deliver to all the customers since the customers' locations is closed.

Table 7 Summary of the result optimal solution

Trip	Number of Vehicle Used	Total Distance (KM)	Path
1	2	1210.92	Path 1: 0 –> 1 –> 0 Path 2: 0 –> 2 –> 3 –> 0
2	2	1776.65	Path 1: 0 –> 1 –> 0 Path 2: 0 –> 4 –> 2 –> 3 –> 0
3	2	1428.54	Path 1: 0 –> 5 –> 0 Path 2: 0 –> 4 –> 3 –> 1 –> 2 –> 0
4	2	2204.81	Path 1: 0 –> 3 –> 0 Path 2: 0 –> 5 –> 4 –> 1 –> 2 –> 0
5	1	707.45	Path 1: 0 –> 1 –> 2 –> 3 –> 0

Table 8 shows the optimal solution for the shortest distance using the ACO algorithm. Relative Percentage Deviation (RPD) has been calculated using the following formula to show any difference or big difference by using the ACO algorithm and before using the ACO algorithm. From the RPD calculation, three of the trips have a reduction after the optimization.

$$\text{RPD} = \frac{\text{Benchmark Distance - ACO Distance}}{\text{Benchmark Distance}} \times 100\% \tag{14}$$

Table 8 Computational result of ACO

Trip	RPD(%)	Distance (km)
1	0	1210.92
2	3.74	1776.65
3	35.80	1428.54
4	11.34	2204.81
5	0	707.15

Some of the data have been changed to show the impact of using vehicles with larger and smaller capacities. The data have been changed for Trip 1, where demand from Trip 1 has decreased from 1200 cartons to 600 cartons, demand at Node 1 in Trip 2 has changed from 1200 cartons to 600 cartons, demand at Node 5 in Trip 3 changed from 1200 cartons to 600 cartons, and demand at Node 3 in Trip 4 change from 1200 cartons to 600 cartons. Trip 5 remains unchanged because it already gives the optimal solution and will be the same even though some of variable is changed and the changes is to decrease the demand, even though the demand decreases or not in Trip 5, the Trip still can be handled with one vehicle and will not change the result. The capacity of the vehicle changed to 2000 for increasing and 600 for decreasing the vehicle capacity.

Table 9 Result when the capacity is increased

Trip	Number of Vehicles Used	Total Distance (KM)	Path
1	1	983.64	Path 1: 0–>1–> 2 –> 3 –> 0
2	2	1628.96	Path 1: 0 –> 1 –> 2 –> 4–> 0 Path 2: 0 –> 3–> 0
3	2	1428.54	Path 1: 0 –> 2 –> 1 –> 3–> 4–> 0 Path 2: 0 –> 5–> 0
4	2	1944.34	Path 1: 0 –> 1 –> 2–> 3–> 4 –> 0 Path 2: 0 –> 2 –> 5 –> 0
5	1	707.45	Path 1: 0 –> 1 –> 2 –> 3–> 0

Results in Table 9 show the proven that when the demand is small or decreases and the vehicle with the largest capacity is used, may lead to fewer numbers and path routes being used. Thus, the total distance will be decreased. For Trip 3, the total distance is the same even though the path selection has changed because the result is already giving the best short distance even some of the variables have been changed and the selected path already satisfies the constraint. Trip 5 is the same because the variable remains unchanged.

To show the effect when the capacity of the vehicle is small, the capacity of the vehicle has been decreased and changed to a maximum 600 cartons.

Enhanced Knowledge in Sciences and Technology Vol. 4 No. 2 (2024) p. 45-55

Table 10 Result when the capacity is decreased

Trip	Number of Vehicles Used	Total Distance (KM)	Path
1	3	1352.06	Path 1: 0 –>1 –> 0 Path 2: 0 –> 2 –> 0 Path 3: 0 –> 3 –> 0
2	4	2802.69	Path 1: 0 –> 3 –> 0 Path 2: 0 –> 1 –> 0 Path 3: 0 –> 4 –> 0 Path 4: 0 –> 2 –> 0
3	3	1946.36	Path 1: 0 –> 5 –> 0 Path 2: 0 –> 3 –> 4 –> 0 Path 3: 0 –> 2 –> 1 –> 0
4	3	2708.40	Path 1: 0 –>4 –> 5 –> 0 Path 2: 0 –>2 –> 1 –> 0 Path 3: 0 –> 3 –> 0
5	3	1384.84	Path 1: 0 –> 2 –> 0 Path 2: 0 –> 3 –> 0 Path 3: 0 –> 1 –> 0

Table 10 shows the result when the capacity of the vehicle is small. It shows that Trip 1, Trip 2, Trip 3, Trip 4 and Trip 5 does not give a better short distance compared to when the vehicle has a large capacity. The results in Tables 9 and 10 show that using the largest vehicle capacity can improve the finding of the shortest distance for delivery.

Table 11 demonstrates the impact of the broad time windows. The time windows have been changed and extended from 1 day to 2 days, and the maximum vehicle capacity has changed from 1200 cartons to 2000 cartons. The outcomes reveal that employing vehicles with larger capacities alongside broad time windows significantly enhances the ability to optimize delivery distances.

Table 11 Result when have the broad time windows

Trip	Number of Vehicles Used	Total Distance (KM)	Path
1	2	1124.79	Path 1: 0 –> 1 –> 2 –> 0 Path 2: 0 –> 3 –> 0
2	2	1628.96	Path 1: 0 –> 1 –> 2 –> 4 –> 0 Path 2: 0 –> 3 –> 0
3	2	1428.54	Path 1: 0 –> 2 –> 1 –> 3 –> 4 –> 0 Path 2: 0 –> 5 –> 0
4	2	1783.43	Path 1: 0 –> 1 –> 3 –> 2 –> 0 Path 2: 0 –> 4 –> 5 –> 0
5	1	707.45	Path 1: 0 –> 1 –> 2 –> 3 –> 0

For Trip 3, the total distance is the same even though the result already gives the best short distance, even though some of the variables have been changed and the selected path already satisfies the constraint. Trip 5 is also remains unchanged because there’s no other best solution because it already has the best short distance even if we extend the time windows since the place of the customers is closed.

4. Conclusion

Distance and optimum routing are important factors in determining the best solution for scheduling the delivery path. This study implemented the Ant Colony Algorithm to solve a capacitated vehicle routing problem with time windows for scheduling the delivery to the customer. The results are obtained with the assistance of Python. We had achieved all the objectives of this study. The first is to determine the optimal solution for the Capacitated Vehicle Routing Problem with time windows for fleet management using Ant Colony algorithms. The 14 locations of the customers, the demands and the time windows were obtained from the soy sauce manufacturing industry. To achieve this objective, we ran the program in Python software 10 times for every trip. The location of the 14 customers included the warehouse of the soy sauce manufacturing industry located at Kempas lama, Johor while the customers is located at Muar, Segamat, Gong Badak, Ipoh, Semenyih, Mentakab, Kota Bharu, Muar, Ayer Hitam, Seremban, Kuala Selangor, Kuantan, Bukit Bendera, and Tanjung Minyak. The result of the optimal solution to find the best shortest distance for delivery obtained with 1210.92 km, 1776.65 km, 1428.54 km, 2204.81 km, and 707.45 km for Trip 1, Trip 2, Trip 3, Trip 4, and Trip 5, respectively. This solution aims to fulfill the secondary objective of proposing efficient short distances for vehicle routing in fleet management transportation. To have efficient vehicle routing in fleet management transportation, we propose the result of the optimal solution as the best short distance to make the delivery. While, to make the delivery more efficient, we propose to have a large capacity of the fleet vehicle and broad time windows for the delivery.

Acknowledgement

The authors would 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. No financial or personal relationship with individuals or organizations has inappropriately influenced our work.

Author Contribution

*The author confirms contribution to the paper as follows: **study conception and design**: Muhammad Afiq, Siti Suhana Jamaian; **analysis and interpretation of result**: Muhammad Afiq; **validation of results**: Siti Suhana Jamaian; **draft manuscript preparation**: Muhammad Afiq. All authors reviewed the results and approved the final version of the manuscript. They have agreed to be accountable for all aspects of the work, ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.*

References

[1] Godfrey, G. A., and Powell, W. B. (2002). An Adaptive dynamic Programming Algorithm for Dynamic Fleet Management, i: Single Period Travel Times. Transportation Science, 36(1), 21–39. https://doi.org/10.1287/trsc.36.1.21.570

[2] MarketsAndMarkets (2023). Fleet Management Market worth $52.4 billion by 2027, growing at a CAGR of 15.5%. Retrieved on April 13, 2023, from https://www.globenewswire.com/newsrelease/2023/04/13/2646380/0/en/Fleet-Management-Market-worth-52-4-billion-by-2027-growing-at-a-CAGR-of- 15-5-Report-by-MarketsandMarkets.html.

[3] Islam, R., and Rahman, M. S. (2012). An ant colony optimization algorithm for waste collection vehicle routing with time windows, driver rest period and multiple disposal facilities. In *2012 International Conference on Informatics, Electronics & Vision (ICIEV)* pp.774 779.https//doi.org/10.1109/ICIEV.2012.6317421

[4] Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A., L., A. (2013). review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), pp.1-11. https://doi.org/10.1016/j.ejor.2012.08.015

[5] Dantzig, G. B., and Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1), pp. 80-91. https://doi.org/10.1287/mnsc.6.1.80

[6] Lysgaard, J., Letchford, A. N., & Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming, 100(2), 423–445. https://doi.org/10.1007/s10107-003-0481-8

[7] Truden, C., Maier, K., and Armbrust, P. (2022). Decomposition of the vehicle routing problem with time windows on the time dimension. *Transportation Research Procedia*, 62, pp.131-138. https://doi.org/10.1016/j.trpro.2022.02.017

[8] Ashima, G, and Saini. S. (2017) An Enhanced Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows. Ninth International Conference on Advanced Computing (ICoAC), 2017 pp.267-274. 10.1109/ICoAC.2017.8441175

[9] Li, X. (2015). Capacitated vehicle routing problem with time windows: a case study on pickup of dietary products in nonprofit organization*. Arizona State University,* pp.1-71.

[10] Razavi, M., and Eshlaghy, A. T. (2015). Using an ant colony approach for solving capacitated vehicle routing problem with time windows. *Research Journal of Recent Sciences ISSN*, *2277*, 2502. pp.30-35.