Sinergi International Journal of Logistics
E-ISSN: 2988-6244
Volume. 2, Issue 1, February 2024
Page No. 17-29

Capacitated Vehicle Routing Problem (CVRP) with Sweep and Nearest Neighbor Algorithm

Erly Ekayanti Rosyida[1], Sugianto[2], Imaduddin Bahtiar Efendi[3]
[123] Universitas Islam Majapahit, Indonesia
Correspondent : erlyekayantir@gmail.com[1]

Received : January 2, 2024
Accepted : February 12, 2024
Published : February 29, 2024

Citation: Rosyida, E, E., Sugianto., Efendi, I, B. (2024). Capacitated Vehicle Routing Problem (CVRP) with Sweep and Nearest Neighbor Algorithm. Sinergi International Journal of Logistics, 2(1), 17-29.

ABSTRACT: The Capacitated Vehicle Routing Problem (CVRP) presents significant challenges in shipping route optimization and logistics management. These challenges include balancing vehicle capacity, minimizing travel distance, and efficiently grouping delivery points, all of which are crucial for enhancing operational efficiency and reducing costs. This research aims to apply a combination of the Sweep and Nearest Neighbor algorithms to address the CVRP, seeking to improve route efficiency and manage vehicle capacity effectively. The Sweep algorithm is employed to cluster pickup points based on their polar angle from the depot, facilitating efficient grouping and optimal vehicle capacity management. Within each cluster, the Nearest Neighbor algorithm is implemented to optimize the sequence of visits, minimizing total travel distance by sequentially selecting the next closest point. The Haversine Distance is used to calculate the distances between points, ensuring geographical accuracy compared to the Euclidean method. Experimental results demonstrate that this hybrid approach yields shorter routes. Quantitative analysis shows a significant reduction 13% in total travel distance when using this combination of algorithms, highlighting its effectiveness in solving the CVRP. This research demonstrates that combining the Sweep and Nearest Neighbor algorithms provides an efficient solution to the CVRP, improving route optimization and vehicle capacity management. The findings contribute valuable insights to logistics management, with practical implications for enhancing shipping route efficiency.

Keywords: CVRP, Sweep Algorithm, Nearest Neighbor, Optimization

INTRODUCTION

In today's era of globalization and rapid economic development, optimizing logistics and goods distribution systems has become crucial for maintaining competitive advantage. Effective delivery route planning is essential for reducing operational costs, enhancing delivery speed, and ensuring high levels of customer satisfaction. Efficient route optimization not only minimizes costs but also significantly improves service quality, making it a vital component of logistics management (Avraham et al., 2017; Laganà et al., 2015; Liu & Lin, 2019; Seifbarghy & Samadi, 2014).

Planning delivery routes involves various complex challenges that must be addressed to achieve maximum efficiency (Bertrand et al., 1986; Scharpff et al., 2021; Zhou et al., 2019). Ensuring that each vehicle's load does not exceed its capacity while meeting all customer demands, minimizing the total travel distance or time, which directly impacts operational costs, reducing costs associated with fuel, labor, and vehicle maintenance and variability in demand, which can significantly fluctuate in both quantity and location, requires flexibility and adaptability in route planning (Das, 2016; Evans et al., 2011; Roudo et al., 2018). These challenges serve as a reference in planning delivery strategies that provide the most optimal conditions (Ghannadpour et al., 2014; Zhao et al., 2021). The approach used to solve this problem is the Capacitated Vehicle Routing Problem (CVRP), a variant of the Vehicle Routing Problem (VRP).

The Capacitated Vehicle Routing Problem (CVRP) is a crucial optimization challenge in logistics and supply chain management. It involves planning optimal routes for a fleet of vehicles with limited capacity to serve customers while minimizing total travel distance (Fischetti et al., 1994; Rojas-Cuevas et al., 2018). Efficiency is achieved by strategizing routes for each fleet to meet all customer demands, thereby optimizing the number of fleets assigned (Fischetti et al., 1994). Improved routing can enhance delivery speed and reliability. For instance, companies like UPS have reported reducing their delivery times by 10-15% through optimized routing strategies, leading to improved customer satisfaction and service levels. Beside that, proper route optimization ensures that each vehicle operates at its full capacity, thereby reducing the number of vehicle.

Various solution approaches have been developed, including classical heuristics, metaheuristics, and exact methods (Konstantakopoulos et al., 2022). They also explain about the common strategy, split the task into two phases: customer clustering and route optimization. The Sweep algorithm is popular for clustering, with recent improvements focusing on identifying appropriate starting angles ((Peya et al., 2019). The Sweep algorithm is a heuristic method used to solve vehicle routing problems by utilizing geometry and angles to cluster customers into manageable routes for a limited capacity fleet (Akhand et al., 2017; Peya et al., 2019). The basic Sweep Algorithm clusters nodes based on polar angles, but variations have been proposed to improve performance. Peya et al. (2019) explored different starting angles for clustering for route optimization. These modifications can lead to better solutions for CVRPs compared to conventional approaches. The algorithm's effectiveness is influenced by problem instance characteristics, such as node distribution and depot location.

While the Sweep and Nearest Neighbor algorithms are frequently used to address the Capacitated Vehicle Routing Problem (CVRP), existing implementations often rely on Euclidean distance, which can introduce inaccuracies in distance calculations due to the Earth's curvature. The Haversine formula is a well-established method for calculating distances between two points on a spherical surface and is particularly useful for geographic applications ((Prasetya et al., 2020; Winarno et al., 2017). It has been applied in various contexts, including location-based services for presence systems (Winarno et al., 2017) and route optimization (Prasetya et al., 2020). By integrating Haversine distance into these algorithms, this study aims to improve the efficiency of route optimization, offering more realistic and applicable solutions for real-world geographical contexts by explore a combination of the Sweep and Nearest Neighbor algorithms, enhanced with more efficient distance-based clustering techniques, to improve accuracy and efficiency in CVRP solutions under dynamic demand. We will also compare the results with those obtained using the Sweep algorithm and the Nearest Neighbor algorithm individually (McDaniel et al., 2023; Prajapati et al., 2022; Wang et al., 2018).

The remainder of the paper is structured as follows: Section II provides a detailed explanation of the algorithm including a brief overview of the Sweep, Nearest Neighbor, combine of Sweep-Nearest Neighbor algorithms and haversine distance method for better clarity. Section III presents data analyses and a comparison of different techniques applied to benchmark CVRPs. Finally, Section IV summarizes the conclusions of the paper.

METHOD

The purpose of this investigation is to analyze the performance of the Sweep and Nearest Neighbor algorithms in solving the Capacitated Vehicle Routing Problem (CVRP) amid varying demand. This section begins by outlining the haversine distance, sweep algorithm and Nearest Neighbor algorithm (Ali et al., 2023; Mahajan et al., 2019; Mengash et al., 2023).

Haversine Distance

The Haversine distance is a formula used to calculate the great-circle distance between two points on the surface of a sphere, given their longitudes and latitudes. This method is particularly useful for determining the distance between geographical locations on Earth. This method calculates the shortest distance over the Earth's surface, giving a result in a straight line distance between two points (as the crow flies). It provides a more accurate distance than Euclidean distance for points on the Earth's surface. Beside that, it easy to implement with basic trigonometric functions. The Haversine Distance equation is as follows:

$$a = \mathrm{Sin}^2(\Delta x/2) + \mathrm{Cos}(x_1) * \mathrm{Cos}(x_2) * \mathrm{Sin}^2(\Delta y/2) \tag{1}$$

$$c = 2 * \mathrm{atan2}(\sqrt{a}, \sqrt{1-a}) \tag{2}$$

$$d = R * c \tag{3}$$

$$radians = degree * \frac{\pi}{180} \tag{4}$$

Where x_1 and x_2 are the latitudes of the two points radians. To convert the latitudes and longitudes degree to radians use function 4. Δx is difference in latitudes $(x_2 - x_1)$ and Δy is difference in longitudes $(y_2 - y_1)$. R is the earth's radius (mean radius = 6,371 km) and d is the distance between two points. Algorithm 1 shows the steps of haversine distance.

Sweep Algorithm

Sweep algorithm is a specific heuristic approach used to generate feasible routes for vehicles in variations of the Vehicle Routing Problem (VRP). It operates by creating routes based on angular sectors around a central depot. The fundamental idea is to group customers into clusters according to their polar angles relative to the depot and then assign each cluster to a vehicle. The Sweep Algorithm is used to route customers based on their polar angle relative to the depot. The polar angle Equation given by:

$$\theta_i = \mathrm{atan2}\,(y_i - y_0, x_i - x_0) \tag{5}$$

Where i is customer index $(i = 1,2,3,4, \dots\dots, n)$ and θ_i is customer i polar angle. (x_0, y_0) is the depot location and (x_i, y_i) is the customer location. Convert polar angle into degree by this equation:

$$\theta_i \; in\; degree = \theta_i \; in\; radian * \frac{180}{\pi} \tag{6}$$

The steps of the sweep algorithm illustrated in Algorithm 2.

Nearest Neighbor

The Nearest Neighbor Algorithm is a method used to find the closest point or points to a given query point from a set of points. In VRP context, nearest neighbor is a heuristic approach where the algorithm starts at an initial point, moves to the nearest unvisited point at each step, and continues until it has visited all points and returned to the starting point. This approach does not always provide the optimal solution but can be useful for approximations.

Sweep Nearest Neighbor Algorithm

The Sweep Nearest Neighbor Algorithm combines the nearest neighbor approach with a sweeping mechanism to reduce the search space and improve efficiency. It's particularly useful for problems where the spatial distribution of points can be exploited to find better solutions more quickly. The Sweep algorithm is used to route based on the polar angle of each customer. While the Nearest Neighbor algorithm is used to optimize the results obtained from the Sweep algorithm. The output of the Sweep algorithm consists of customer groupings based on their polar angle order for each fleet. The routes formed for each fleet are still based on their polar angle order. This result is believed to be further optimized by finding the shortest route using Nearest Neighbor algorithm.

Algorithm 1: Haversine distance

Step 1: Convert the latitude and longitude of both points from degrees to radians.

Step 2: Calculate the differences in latitude and longitude

Step 3: Apply the Haversine formula using Equation 1

Step 4: Calculate the central angle using Equation 2

Step 5: Calculate the distance using Equation 3

Algorithm 2: Sweep Algorithm

Initialization:

Step 1: Specify the problem parameters, such as the locations of customers and the depot, customer demands, and vehicle capacities.

Step 2: Calculate the distance with haversine distance algorithm

Step 3: Calculate Polar angles of each customer using Equation 5.

Step 4: Convert the polar angles from radians to degrees for easier interpretation and sorting use equation 6.

Step 5: Sort the customers based on their polar angles in ascending order

Clustering:

Step 1: Set $cluster = 1$

Step 2: Start from the smallest polar angle and move to the largest add them to the current cluster.

Step 3: Stop adding customer when they cannot be added to the current cluster because exceeding vehicle capacity

Step 4: Set $cluster = cluster + 1$

Step 5: Repeat step 2-4 until each customer have been allocated

Step 6: Construct routes based on the results of the clustering. Add the depot as both the starting and ending point for each route.

Step 7: Calculate the total distance for each route.

Algorithm 3: Nearest Neighbor Algorithm

Initialization:

Step 1: Specify the problem parameters, such as the locations of customers and the depot, customer demands, and vehicle capacities.

Step 2: Calculate the distance with haversine distance algorithm

Step 3: Set $route = 1$

Step 4: set the depot as starting and end point then find the nearest customer from starting point and add the nearest customer to the route

Step 5: Update the vehicle's load by adding the demand of the selected customer then mark the customer as visited and move the vehicle to this customer's location.

Step 6: Continue selecting the nearest customer and adding it to the route

Step 7: Stop adding customer when they cannot be added to the current route because exceeding vehicle capacity

Step 8: Set $route = route + 1$

Step 9: Repeat step 2-5 until each customer have been allocated

Step 10: Ensure all customers are visited and all vehicles return to the depot then calculate the total distance for each route.

Algorithm 3: Sweep-Nearest Neighbor Algorithm

Initialization:

Step 1: Specify the problem parameters, such as the locations of customers and the depot, customer demands, and vehicle capacities.

Step 2: Calculate the distance with haversine distance algorithm

Step 3: Calculate Polar angles of each customer using Equation 5.

Step 4: Convert the polar angles from radians to degrees for easier interpretation and sorting use equation 6.

Step 5: Sort the customers based on their polar angles in ascending order

Clustering:

Step 1: Set $cluster = 1$

Step 2: Start from the smallest polar angle and move to the largest add them to the current cluster.

Step 3: Stop adding customer when they cannot be added to the current cluster because exceeding vehicle capacity

Step 4: Set $cluster = cluster + 1$

Step 5: Repeat step 2-4 until each customer have been allocated

Routing:

Step 1: For each cluster

Step 2: Set the depot as both the starting and ending point of the route

Step 3: find the nearest customer from starting point and add the nearest customer to the route. Update the vehicle's load by adding the demand of the selected customer then mark the customer as visited and move the vehicle to this customer's location

Step 4: Continue selecting the nearest customer and adding them to the route until all customers in the cluster have been visited

Step 5: Repeat step 2-4 until each cluster have been optimized

Step 6: Calculate the total distance for each route.

RESULT AND DISCUSSION

This section explain the primary data and data analysis setup. Then it describes the data analysis and discuss about the comparison of the result with highlight similarities, differences, and any new insights provided.

This study uses primary data, including a number of customer with their order quantities and coordinat location. Then vehicle capacity. The CVRP approach is used to plan delivery strategies by determining routes that provide an optimal total delivery distance. Route determination is conducted for a mineral water company that will distribute to 30 customers, with the company serving as the depot. The fleet used for this delivery has a capacity of 490 units per fleet. This fleet is used to transport three types of water products, all of which are of the same size. The customer data, shown in Table 1, includes the locations of the depot and customers, as well as the demand for each customer.

This study was conducted using Python, with Google Colab serving as the platform for executing code and managing computational tasks. Python's libraries and functions facilitate the implementation of the Sweep and Nearest Neighbor algorithms, allowing for efficient data processing and route optimization.

Table 1. The Primary Data

Customer	Coordinate		Demand	Customer	Coordinate		Demand
	x	y	(unit)		x	y	(unit)
Depot	-7,49601	112,46659	-				
Customer 1	-7,33597	112,72626	100	Customer 16	-7,24157	112,60967	102
Customer 2	-7,25996	112,68108	89	Customer 17	-7,37378	112,96307	96
Customer 3	-7,33165	112,67044	98	Customer 18	-7,36452	112,68756	100
Customer 4	-7,26749	112,76099	104	Customer 19	-7,39964	112,72361	100
Customer 5	-7,25567	112,66393	88	Customer 20	-7,39659	112,69882	95
Customer 6	-7,30212	112,72909	102	Customer 21	-7,35672	112,75370	102
Customer 7	-7,31649	112,79734	97	Customer 22	-7,50769	112,71182	97
Customer 8	-7,29126	112,79241	88	Customer 23	-7,42706	112,67333	86

Customer 9	-7,27687	112,75092	105	Customer 24	-7,46338	112,57118	102
Customer 10	-7,30540	112,73096	101	Customer 25	-7,45017	112,72938	103
Customer 11	-7,30083	112,73216	88	Customer 26	-7,36396	112,58110	90
Customer 12	-7,33951	112,75206	97	Customer 27	-7,45441	112,67694	96
Customer 13	-7,28292	112,72193	94	Customer 28	-7,40742	112,58110	95
Customer 14	-7,26964	112,72319	102	Customer 29	-7,46480	112,71529	99
Customer 15	-7,25149	112,64408	114	Customer 30	-7,34746	112,68718	100

In the data analysis section, delivery routes are determined using three types of algorithms: Sweep, Nearest Neighbor, and a combination of Sweep and Nearest Neighbor. The distances for all three algorithms are computed using the Haversine Distance algorithm. Table 2 presents the distance calculations. The distance matrix, calculated using the Haversine distance algorithm (Algorithm 1), determines the distances between each point (depot and all customers). Table 3 illustrates the results of the polar angle calculations, which are performed using Equations 5 and 6. Equation 5 calculates the polar angle in radians, and Equation 6 is used to convert the polar angle from radians to degrees.

The clustering process results are presented in Table 4, showing the outcomes of sorting based on the magnitude of the polar angles. The initial order starts with the customer with the smallest polar angle. Customers are then plotted and grouped into clusters (fleets) based on this order, with grouping constrained by fleet capacity. Seven clusters/vehicles are assigned, with each load remaining under the vehicle's capacity.

Table 5 shows the results of optimization using the three algorithms: Sweep, Nearest Neighbor, and the combination of Sweep and Nearest Neighbor. The results include vehicle assignments, routes for each vehicle with the total distance, and the total distance for all fleets for each algorithm. The comparison is based on the total distance for each algorithm start with customer

The results show the similarities and differences in the outcomes of applying three algorithms to CVRP. **The similarity is that the number of clusters or fleets assigned is 7. On the other hand, the differences are in the routes and visit sequences. The Sweep algorithm and the Sweep-Nearest Neighbor algorithm** produced the same clustering results in the analysis but generated different routes. In the routing optimization process, the Sweep algorithm bases its optimization on the size of the polar angle for each customer (Algorithm 2). In contrast, the Sweep-Nearest Neighbor algorithm is based on the proximity customer's location within the assigned clusters (Algorithm 4). The total distance calculated by the Sweep-Nearest Neighbor algorithm is 591,97, compared to 680,89 for the Sweep algorithm, resulting in a 13% difference.

For better understanding, refer to Figure 1 and Figure 3. Figure 1 illustrated the generated route of Sweep algorithm, while Figure 3 explained the route of combine algorithm (Sweep and Nearest Neighbor). It is noteworthy from the figure that the better result route sequence is came from the combination of Sweep and Nearest Neighbor algorithm. Fleet 1 in Sweep algorithm resulted 0-16-15-5-26-2-0, while fleet 1 in Sweep-Nearest Neighbor algorithm resulted 0-26-16-15-5-2-0. The sequence of visits for Fleet 1 illustrated in the figure shows that Customer 26 was visited first in the Sweep-Nearest Neighbor algorithm because its location is closest to the depot. In contrast, Customer 16 was visited first in the Sweep algorithm because it has the smallest polar angle. Then in the Sweep algorithm, the next visit after Customer 5 is Customer 26, even though Customer 2 is closer to Customer 5, but has a larger polar angle than Customer 26. In contrast, the Sweep-Nearest Neighbor algorithm visits the nearest customer after Customer 5, which is Customer 2. Based on the data analysis, the size of the polar angle does not correlate directly with the distance. This results in a larger distance in the visit sequence for the Sweep algorithm.

Table 3. Polar Angle Results

Criteria	Node	Polar Angle (radian)	Polar Angle (Degree)	Criteria	Node	Polar Angle (radian)	Polar Angle (Degree)
Customer 1	1	1,02	58,38	Customer 16	16	0,51	29,37
Customer 2	2	0,74	42,28	Customer 17	17	1,33	76,21
Customer 3	3	0,89	51,15	Customer 18	18	1,03	59,27
Customer 4	4	0,91	52,21	Customer 19	19	1,21	69,48
Customer 5	5	0,69	39,41	Customer 20	20	1,17	66,86
Customer 6	6	0,93	53,58	Customer 21	21	1,12	64,15
Customer 7	7	1,07	61,54	Customer 22	22	1,62	92,77
Customer 8	8	1,01	57,88	Customer 23	23	1,25	71,59
Customer 9	9	0,91	52,40	Customer 24	24	1,27	72,71
Customer 10	10	0,95	54,24	Customer 25	25	1,40	80,15
Customer 11	11	0,94	53,71	Customer 26	26	0,71	40,95
Customer 12	12	1,07	61,30	Customer 27	27	1,38	78,85
Customer 13	13	0,88	50,18	Customer 28	28	0,91	52,30
Customer 14	14	0,85	48,61	Customer 29	29	1,45	82,89
Customer 15	15	0,63	35,99	Customer 30	30	0,98	56,07

Meanwhile, the Nearest Neighbor algorithm results in a larger total distance compared to the Sweep-Nearest Neighbor algorithm. The clustering results from the Nearest Neighbor algorithm differ from those of the Sweep algorithm and Sweep-Nearest Neighbor algorithms. In the Nearest Neighbor algorithm, Cluster or Fleet 1 results in the sequence 0-24-28-26-3-30-0. In Figure 2, for Cluster or Fleet 1, the first customer selected is based on proximity to the depot (Customer 24), while in the other two algorithms, the first customer in Cluster 1 is the one with the smallest polar angle (Customer 16). This condition leads to different routing outcomes for each fleet. However, the Sweep-Nearest Neighbor algorithm produces a better total distance, with a 3% improvement over the Nearest Neighbor algorithm

Tabel 4. Customer Clustering Results

Customer	Polar Angle (Degree)	Demand	Demant Cumulative	Fleet	Customer	Polar Angle (Degree)	Demand	Demand Cumulative	Fleet
Customer 16	29,37	102	102		Customer 12	61,30	97	97	
Customer 15	35,99	114	216		Customer 7	61,54	97	194	Fleet 5
Customer 5	39,41	88	304	Fleet 1	Customer 21	64,15	102	296	
Customer 26	40,95	90	394		Customer 20	66,86	95	391	
Customer 2	42,28	89	483		Customer 19	69,48	100	100	Fleet 6
Customer 14	48,61	102	102	Fleet 2	Customer 23	71,59	86	186	

Customer 13	50,18	94	196	
Customer 3	51,15	98	294	
Customer 4	52,21	104	398	
Customer 28	52,30	95	95	Fleet 3
Customer 9	52,40	105	200	
Customer 6	53,58	102	302	
Customer 11	53,71	88	390	
Customer 10	54,24	101	101	Fleet 4
Customer 30	56,07	100	201	
Customer 8	57,88	88	289	
Customer 1	58,38	100	389	
Customer 18	59,27	100	489	
Customer 24	72,71	102	288	
Customer 17	76,21	96	384	
Customer 27	78,85	96	480	
Customer 25	80,15	103	103	Fleet 7
Customer 29	82,89	99	202	
Customer 22	92,77	97	299	

Table 5. CVRP Results with Sweep, Nearest Neighbour and Sweep-Nearest Neighbor Algorithm

Fleet	Sweep		Nearest Neighbor		Sweep-Nearest Neighbor	
	Route	Total Distance (Km)	Route	Total Distance (Km)	Route	Total Distance (Km)
Fleet 1	0-16-15-5-26-2-0	105,01	0-24-28-26-3-30-0	65,67	0-26-16-15-5-2-0	76,81
Fleet 2	0-14-13-3-4-0	100,70	0-27-23-20-19-18-0	67,65	0-3-13-14-4-0	83,71
Fleet 3	0-28-9-6-11-0	80,23	0-22-29-25-21-11-0	87,93	0-28-6-11-9-0	79,55
Fleet 4	0-10-30-8-1-18-0	98,51	0-16-15-5-2-13-0	82,51	0-18-30-1-10-8-0	87,74
Fleet 5	0-12-7-21-20-0	83,54	0-1-12-10-6-8-0	91,07	0-20-21-12-7-0	84,50
Fleet 6	0-19-23-24-17-27-0	149,38	0-14-9-4-7-0	90,89	0-24-27-23-19-17-0	116,15
Fleet 7	0-25-29-2-0	63,52	0-17-0	112,81	0-22-29-25-0	63,52
Total Distance		**680,89**	**Total Distance**	**598,77**	**Total Distance**	**591,97**

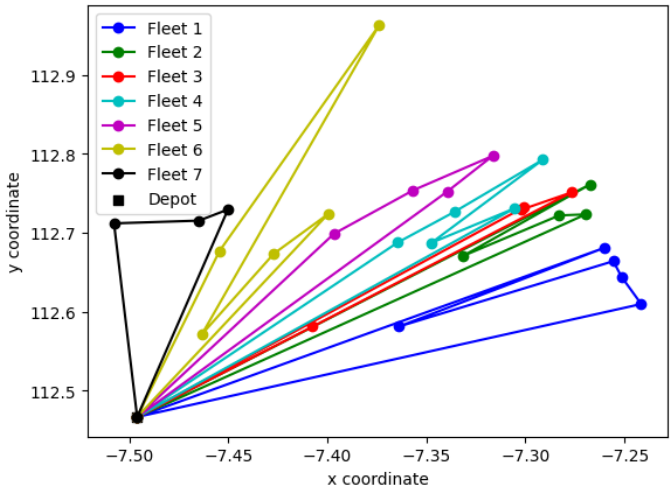

Figure 1. Solution with Sweep Algorithm

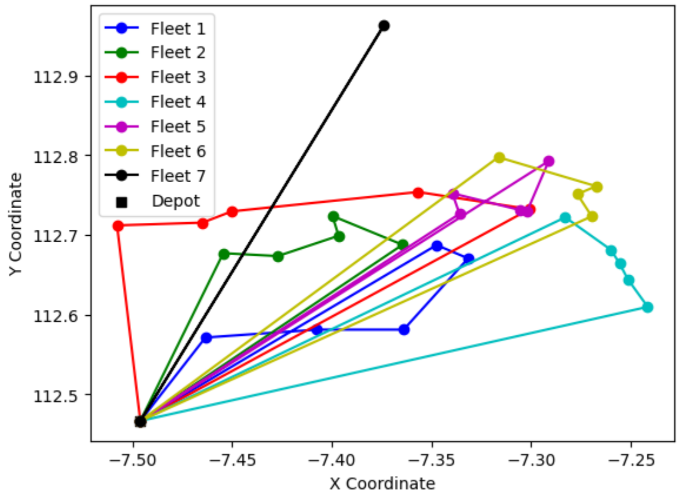

Figure 2. Solution with Nearest Neighbor Algorithm

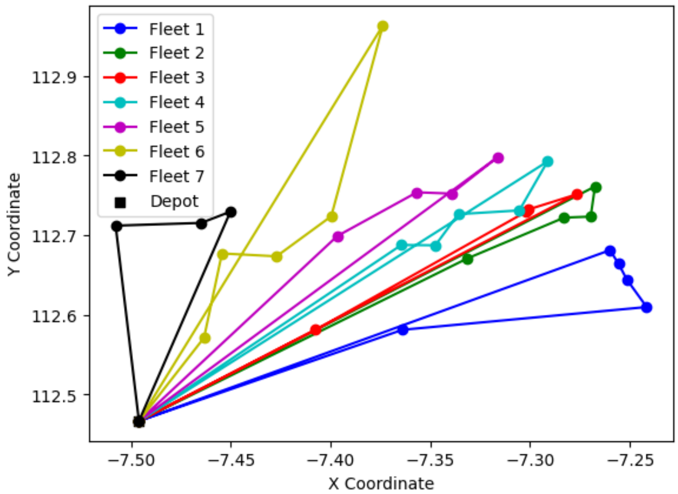

Figure 3. Solution with Sweep-Nearest Neighbor Algorithm

CONCLUSION

This study compared the performance of three algorithms: Sweep, Nearest Neighbor, and the Sweep-Nearest Neighbor algorithm, for solving the Capacitated Vehicle Routing Problem (CVRP). The comparison among these algorithms is based on the total distance results, with distances between locations calculated using the Haversine Distance Algorithm. The analysis revealed that while the Sweep algorithm produced the largest total distance, the Sweep-Nearest Neighbor algorithm offered a more efficient routing sequence by focusing on the nearest distance between locations, leading to a more optimal total distance compared to the Nearest Neighbor algorithm.

The Sweep-Nearest Neighbor algorithm, which integrates both the Sweep and Nearest Neighbor approaches, demonstrated superior performance by optimizing both clustering and route sequencing. This method resulted in the shortest total distance and showed a 3% improvement over the Nearest Neighbor algorithm and a 13% improvement over the Sweep algorithm. Overall, the Sweep-Nearest Neighbor algorithm emerged as the most effective for minimizing total travel distance while maintaining practical efficiency in route planning. This underscores the advantage of combining heuristic and nearest neighbor techniques to address complex routing challenges in CVRP.

Although this combination has yielded promising results, further research could explore integrating other optimization algorithms, such as Genetic Algorithms or Simulated Annealing, to achieve even more optimal solutions. Additionally, it is important to consider disruptions affecting the distribution process. Incorporating factors such as customer disruptions, real-time traffic conditions, or a combination of both can enhance distribution planning and improve the accuracy and reliability of the routing system (Rosyida et al., 2020).

REFERENCE

Akhand, M. A. H., Peya, Z. J., Sultana, T., & Rahman, M. M. H. (2017). Solving capacitated vehicle routing problem using variant sweep and swarm intelligence. *Journal of Applied Science and Engineering*, *20*(4), 511–524. https://doi.org/10.6180/jase.2017.20.4.13

Ali, A. M., Ngadi, M. A., Sham, R., & Al_Barazanchi, I. I. (2023). Enhanced QoS Routing Protocol for an Unmanned Ground Vehicle, Based on the ACO Approach. *Sensors*, *23*(3). https://doi.org/10.3390/s23031431

Avraham, E., Raviv, T., & Khmelnitsky, E. (2017). The decentralized field service routing problem. *Transportation Research Part B: Methodological*, *104*, 290–316. https://doi.org/10.1016/j.trb.2017.07.005

Bertrand, J. T., Mangani, N., & Mansilu, M. (1986). Strategies for family planning service delivery in Bas Zaire. *International Family Planning Perspectives*, *12*(4), 108–115. https://doi.org/10.2307/2947981

Das, P. (2016). Uncharted waters: Navigating new configurations for urban service delivery in India. *Environment and Planning A*, *48*(7), 1354–1373. https://doi.org/10.1177/0308518X16640529

Evans, E., Anglin, M. D., Urada, D., & Yang, J. (2011). Promising practices for delivery of court-supervised substance abuse treatment: Perspectives from six high-performing California counties operating Proposition 36. *Evaluation and Program Planning*, *34*(2), 124–134. https://doi.org/10.1016/j.evalprogplan.2010.09.001

Fischetti, M., Toth, P., & Vigo, D. (1994). A Branch-And-Bound Algorithm for the Capacitated Vehicle Routing Problem on Directed Graphs Author (s): Matteo Fischetti , Paolo Toth and Daniele Vigo Published by : INFORMS Stable URL : http://www.jstor.org/stable/171544 REFERENCES Linked references ar. *Operations Research*, *42*(5), 846–859.

Ghannadpour, S. F., Noori, S., Tavakkoli-Moghaddam, R., & Ghoseiri, K. (2014). A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing Journal*, *14*(PART C), 504–527. https://doi.org/10.1016/j.asoc.2013.08.015

Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. *Operational Research*, *22*(3), 2033–2062. https://doi.org/10.1007/s12351-020-00600-7

Laganà, D., Longo, F., & Santoro, F. (2015). Multi-product inventory-routing problem in the supermarket distribution industry. *International Journal of Food Engineering*, *11*(6), 747–766. https://doi.org/10.1515/ijfe-2015-0052

Liu, G.-S., & Lin, K.-P. (2019). A decision support system of green inventory-routing problem. *Industrial Management and Data Systems*, *119*(1), 89–110. https://doi.org/10.1108/IMDS-11-2017-0533

Mahajan, N., Hegyi, A., Hoogendoorn, S. P., & van Arem, B. (2019). Design analysis of a decentralized equilibrium-routing strategy for intelligent vehicles. *Transportation Research Part C: Emerging Technologies*, *103*, 308–327. https://doi.org/10.1016/j.trc.2019.03.028

McDaniel, E. L., Akwafuo, S., Urbanovsky, J., & Mikler, A. R. (2023). Benchmarking a fast, satisficing vehicle routing algorithm for public health emergency planning and response: “Good Enough for Jazz.” *PeerJ Computer Science*, *9*. https://doi.org/10.7717/peerj-cs.1541

Mengash, H. A., Alqahtani, H., Maray, M., Nour, M. K., Marzouk, R., Al-Hagery, M. A., Mohsen, H., & Duhayyim, M. A. (2023). Coati Optimization-Based Energy Efficient Routing Protocol for Unmanned Aerial Vehicle Communication. *Computers, Materials and Continua*, *75*(3), 4805–4820. https://doi.org/10.32604/cmc.2023.037810

Peya, Z. J., Akhand, M. A. H., Sultana, T., & Hafizur Rahman, M. M. (2019). Distance based Sweep Nearest algorithm to solve Capacitated Vehicle Routing Problem. *International Journal of Advanced Computer Science and Applications*, *10*(10), 259–264. https://doi.org/10.14569/ijacsa.2019.0101036

Prajapati, D., Chan, F. T. S., Daultani, Y., & Pratap, S. (2022). Sustainable vehicle routing of agro-food grains in the e-commerce industry. *International Journal of Production Research*, *60*(24), 7319–7344. https://doi.org/10.1080/00207543.2022.2034192

Prasetya, D. A., Nguyen, P. T., Faizullin, R., Iswanto, I., & Armay, E. F. (2020). Resolving the shortest path problem using the haversine algorithm. *Journal of Critical Reviews*, *7*(1), 62–64. https://doi.org/10.22159/jcr.07.01.11

Rojas-Cuevas, I.-D., Caballero-Morales, S.-O., Martinez-Flores, J.-L., & Mendoza-Vazquez, J.-R. (2018). Capacitated vehicle routing problem model for carriers. *Journal of Transport and Supply Chain Management*, *12*(July), 0–9. https://doi.org/10.4102/jtscm.v12i0.345

Rosyida, E. E., Santosa, B., & Pujawan, I. N. (2020). Freight route planning in intermodal transportation network to deal with combinational disruptions. *Cogent Engineering*, *7*(1). https://doi.org/10.1080/23311916.2020.1805156

Roudo, M., Campbell, A., & Delay, S. (2018). Is decentralisation compatible with the application of performance management? The impacts of minimum service standards on the motivation of local government to improve service delivery in the indonesian decentralised system. *Journal of Regional and City Planning*, *29*(2), 135–155. https://doi.org/10.5614/jrcp.2018.29.2.5

Scharpff, J., Schraven, D., Volker, L., Spaan, M. T. J., & de Weerdt, M. M. (2021). Can multiple contractors self-regulate their joint service delivery? A serious gaming experiment on road maintenance planning. *Construction Management and Economics*, *39*(2), 99–116. https://doi.org/10.1080/01446193.2020.1806336

Seifbarghy, M., & Samadi, Z. (2014). A tabu search-based heuristic for a new capacitated cyclic inventory routing problem. *International Journal of Mathematics in Operational Research*, *6*(4), 491–504. https://doi.org/10.1504/IJMOR.2014.063159

Wang, Y., Peng, S., Assogba, K., Liu, Y., Wang, H., Xu, M., & Wang, Y. (2018). Implementation of cooperation for recycling vehicle routing optimization in two-echelon reverse logistics networks. *Sustainability (Switzerland)*, *10*(5). https://doi.org/10.3390/su10051358

Winarno, E., Hadikurniawati, W., & Rosso, R. N. (2017). Location based service for presence system using haversine method. *Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech 2017*, *2018-January*, 1–4. https://doi.org/10.1109/INNOCIT.2017.8319153

Zhao, P., Liu, F., Guo, Y., Duan, X., & Zhang, Y. (2021). Bi-Objective Optimization for Vehicle Routing Problems with a Mixed Fleet of Conventional and Electric Vehicles and Soft Time Windows. *Journal of Advanced Transportation*, *2021*. https://doi.org/10.1155/2021/9086229

Zhou, W., Oyegoke, A. S., & Sun, M. (2019). Service planning and delivery outcomes of home adaptations for ageing in the UK. *Journal of Housing and the Built Environment*, *34*(2), 365–383. https://doi.org/10.1007/s10901-017-9580-3