International Journal of Industrial Engineering, 30(3), 835-851, 2023

MODIFIED SWEEPING ALGORITHM FOR SOLVING CAPACITATED VEHICLE ROUTING PROBLEM WITH RADIAL CLUSTERED PATTERNS

Lubna Altarawneh[1] and Soongeol Kwon[2,*]

[1]Department of Systems Science and Industrial Engineering
Binghamton University
New York, United States of America

[2]Department of Industrial Engineering
Yonsei University
Seoul, Republic of Korea
[*]Corresponding author's e-mail: sg.kwon@yonsei.ac.kr

This research proposes a modification to the Sweeping Algorithm (SWA) to solve the Capacitated Vehicle Routing Problem (CVRP) for real-world cases where locations are present on radial patterns. The SWA solves CVRP instances efficiently by performing angular sweeping to obtain clusters of locations. However, the SWA does not consider radial distances when clustering the locations, which may cause inefficient clustering when there is a presence of locations on radial patterns. Therefore, this research proposes a systematic approach to solve CVRP instances with apparent radial clusters by considering their angular and radial distances in the clustering phase. This method is configurable to locations' geography and can handle different locations' assignments. The experimental results indicate that the proposed heuristic outperforms SWA and its well-known variant, Sweep Nearest Neighbor (SNN), for the targeted instances designed with radial clusters. The results for CVRP benchmark instances show comparable performance when using the proposed heuristic.

Keywords: Capacitated Vehicle Routing Problem; Cluster-First Route-Second Approach; Sweeping Algorithm; Radial Clustered Patterns.

(Received on September 19, 2022; Accepted on March 9, 2023)

1. INTRODUCTION

Because of the increase in online ordering and delivery services, there has been a need to find the shortest route for vehicles to travel cost-effectively. The problem of finding an optimized route for a set of vehicles leaving the depot, visiting several locations, and returning to the depot is referred to as the Capacitated Vehicle Routing Problem (CVRP). The vehicles have limited capacities, locations have demand to be met, and solution routes must guarantee the satisfaction of demand while not exceeding vehicle capacity (Braekers *et al*., 2016). CVRP has significant importance for many industrial applications, especially in the field of transportation, logistics, and distribution. One of the major concerns in the industry is to reduce the cost of the product. Since transportation accounts for a significant part of the cost, the CVRP can be used to optimize delivery routes and reduce its associated cost (Kalatzantonakis *et al*., 2019), where a small decrease in the traveled distance can result in great savings. The CVRP has been extensively studied in the last few decades, with many methods proposed to solve it because of its practical relevance to real-world problems (Zhang *et al*., 2014). Many methods have been proposed to solve the CVRP and fall into two categories of exact methods that result in optimal routes but are computationally expensive, and heuristic and meta-heuristic methods, which have been widely used because of the good solutions they produce in a short computational time.

Exact methods guarantee optimality by computing all possible solutions and can be categorized into four categories: dynamic programming, set partitioning, branch and bound methods, and branch and cut methods. The approaches to solving CVRP have been dominated by the branch and cut methods applied to solve CVRP (Lysgaard *et al*., 2004). The branch and cut and price methods, which are part of the branch and cut methods, have gained favor in the last decade as researchers solved large CVRPs more quickly by combining the generation of cuts and columns (Fukasawa *et al*., 2006). However, because CVRP is an NP-hard problem, exact methods cannot be used to solve large-scale problems. The largest CVRP instance solved by an exact method consistently has less than 50 customers (Baldacci *et al*., 2004). The number of customers, in other cases, can be much larger. In some instances, there can be more than 100 customers, and the problem needs to be solved using heuristic methods. Heuristic methods fall into three categories: constructive, improvement, and two-phase.

DOI: 10.23055/ijietap.2023.30.3.8595 ISSN 1943-670X

Constructive methods create routes and optimize them at the same time. Clarke and Wright's savings algorithm (CW) is one of the most well-known constructive methods to solve the CVRP (Clarke and Wright, 1964). Several modifications have been proposed to solve the CVRP using CW (Altınel and Öncan, 2005; Pichpibul and Kawtummachai, 2012) and other variants of the Vehicle Routing Problem (VRP) (Anbuudayasankar *et al*., 2012; Cinar *et al*., 2016). Improvement heuristics improve solutions by exchanging routes as in 2-opt or 3-opt heuristics. 2-opt is used for improving routes in the TSP and for constructing routes in CVRP as well (Marinelli *et al*., 2018). 2-opt point exchange has been used to improve routes after initial results are obtained (Suthikarnnarunai, 2008). Two-phase methods have been used extensively in the literature for VRP applications (Zajac, 2018; Comert and Yazgan, 2021), including route-first cluster-second and cluster-first route-second. Route-first cluster-second methods start by decomposing one big initial route into different routes. Beasley (1983) was the first to adopt this approach and stated that the second phase is the standard shortest path problem. Cluster-first route-second heuristics are the most common two-phase methods to solve the CVRP because of their simplicity in decomposing the problem and the short computational time they require. The first phase starts with clustering the locations and assigning each cluster to a vehicle, and constructing initial routes, while the second phase optimizes each vehicle route. Clustering is defined as classifying specific patterns of locations into groups; and is considered vital in solving the CVRP efficiently (Yücenur and Demirel, 2011). One of the popular cluster-first route-second heuristics is the Sweeping Algorithm (SWA), known for the good results it produces when solving the CVRP (Gillett and Miller, 1974). The SWA consists of two main phases; the first phase clusters locations based on their angular distances, where locations with small angular distances are clustered and assigned to the same vehicle, and the second phase optimizes each vehicle route.

The SWA is an effective method for solving CVRP instances and produces good solutions resulting in its wide application in many areas, and is considered in this research because of its simplicity and ease of application. However, the SWA may not work well for real-world problems where radial clusters are present. The SWA clusters locations based on their angular distances where locations with close angular distances are grouped. This may work well for general cases or with cases where locations are close angularly and are not widely separated. On the other hand, when considering special cases of locations present on radial positions, SWA may result in a deteriorated solution. Cases where locations are present on radial positions tend to be common in real-world problems where locations are separated by rivers, mountains, or rings of roads and have not been fully addressed in the literature. In such cases, the SWA creates unnecessarily long-traveled distances by requiring detours to reach all locations in a cluster grouped by their angular distances. Many cities around the world contain residential blocks separated by rings of roads that cause the formation of radial clusters. An example of such cases with rings of roads is present in Shanghai and includes three ring roads, as shown in Figure 1 (Chen and Zhao, 2013). The map shows that clustering the locations present on ring roads by their angular distances would not result in an optimal solution. This is because necessary detours to move from a location on one ring road to another will increase the total distance traveled. In this case, allocating vehicles by ring would result in a better solution.

Figure 1. Map of Shanghai showing radial clusters

The main objective of this research is to develop a systematic approach to solve CVRP instances with clusters on radial positions, where angular and radial distances can be considered in the clustering phase when solving the CVRP. The proposed heuristic addresses the limitations of SWA and is configurable by considering the different geographical natures of the locations. The proposed method focuses on improving the clusters formed in the first phase for the targeted instances to reduce vehicles' traveled distances. While in the second phase, the routes are optimized using an off-the-shelf solver for the Traveling Salesman Problem (TSP). The outline of this paper is summarized as follows: Section 2 reviews the literature on the methods used to solve the CVRP, including SWA. Section 3 explains the problem that will be addressed. Section 4 describes the methodology used. Section 5 discusses the experimental results and compares the proposed method, the SWA, and one of its variants, the Sweep Nearest Neighbor (SNN), in solving CVRP instances. Section 6 provides a conclusion and future work.

2. LITERATURE REVIEW

The SWA is a two-phase heuristic method of cluster-first route-second that was introduced as a way to solve the vehicle dispatch problem (Gillett and Miller, 1974). The SWA has been used since to solve VRP and its variants because of its simplicity. Different methods were used in combination with the SWA to improve its performance. Savitri and Kurniawati (2018) proposed a sweep algorithm and MILP for Vehicle Routing Problems with Time Windows (VRPTW) that reduced the traveled distance and fuel cost. Chen *et al*. (2015) proposed a hybrid two-phase SWA and greedy search to solve CVRP; the first phase consists of clustering locations using SWA where solutions are optimized using Nearest Neighbor (NN). The second phase recombines adjacent clusters to form better clusters. A greedy search is then applied to find the shortest path for vehicles. A greedy search algorithm was used to decrease the travel distances of vehicles after constructing routes in several works (Rattanamanee *et al*., 2020). Teodorović and Pavković (1996) developed a VRP model based on SWA, rules of fuzzy arithmetic, and fuzzy logic for problems with uncertain demand at locations. The results indicate its ability to solve instances efficiently and determine near-optimal routes.

Considerable research proposed modifications to the SWA to improve its performance in solving CVRP instances. A summary of the modifications of SWA is shown in Table 1. Renaud and Boctor (2002) proposed a new sweep-based heuristic to solve fleet size and mix VRP. The modified SWA considered two reference points of the depot and the geometrical center of locations to solve Euclidean problems. For non-Euclidean problems, different orders were defined that resulted in numbering locations based on the order in which they will be visited in the resulting tour. The results show that the modified SWA produced good solutions in terms of cluster formation and outperformed the SWA and its hybrid approaches by reducing the vehicles traveled distances. The weakness of SWA in clustering locations based on angular distances was first pointed out by Na *et al*. (2011). A modification was proposed to tackle this weakness by using NN to determine locations for each cluster after the first location in each cluster is assigned. The SNN was tested on benchmark instances and proved to be effective in solving CVRP and resulted in better performance compared to SWA in terms of the total distance traveled. Akhand *et al*. (2017) proposed an adaptive sweep and investigated appropriate cluster starting angles. The adaptive sweep checks angle differences for the consecutive locations and the distances between the locations as well as between the depot and locations. SWA starting from different angles formed better clusters and provided a better CVRP solution compared to the classical SWA. Peya *et al*. (2018) investigated the ability of an adaptive sweep to solve CVRP by starting to form clusters from the maximum preference values. The preference value is calculated for each consecutive location, and the maximum preference point is taken as a starting angle of cluster formation. Peya *et al*. (2019) proposed a Distance-based Sweep Algorithm (DBSA) to solve the CVRP by considering NN, where cluster formation starts from the farthest node and continues by performing NN. The DBSA showed competitive results in terms of total distance traveled when compared with SWA and SNN.

Different applications of SWA and its modifications in solving VRP and its variants produced good solutions in many studies, where most papers modified the SWA by changing the reference point and modifying the starting angle. Previous research focused on the general case of performing SWA on randomly distributed instances. However, few papers addressed the limitations of SWA in clustering locations based on their angular distances. The method of clustering and assigning vehicles solely by angular distance is not always efficient, especially for real-world problems where radial clustering can be considered and result in a better solution. The proposed method tackles the limitations of SWA and provides a better formation of clusters for instances where radial patterns occur by considering the geographical nature of locations. Previous research did not consider instances that have specific patterns of clusters that are present radially. In the real-world many practical problems have specific patterns of clusters that are present radially because of geographical barriers or ring roads where angular clustering might not be efficient and can result in increased traveled distances. Radial-based clustering can reduce the traveled distances for such cases and has not been studied thoroughly (Tarawneh *et al*., 2020).

Table 1. Summary of SWA modification studies

Paper	Description
Gillet and Miller (1974)	Introduced SWA to solve the vehicle dispatch problem by dividing it into two sub-problems
Renaud and Boctor (2002)	Modified the SWA by changing the reference point for Euclidean problems
Na *et al*. (2011)	Modified the SWA by using NNA to assign locations to the location with the smallest polar angle
Akhand *et al*. (2017)	Modified the SWA by considering an adaptive sweep cluster to form clusters based on appropriate cluster starting angles
Peya et. Al (2018)	Modified the SWA by considering a starting point based on the maximum preference point
Zahrul *et al*. (2019)	Modified the SWA by considering a distance-based sweep nearest neighbor that starts clustering from the farthest location and continues for a cluster based on the nearest neighbor concept

3. PROBLEM DESCRIPTION

The CVRP may be defined by a fleet of M vehicles leaving from an initial location, the depot d, denoted by (0,0), where vehicles have limited capacities of Q and locations have demand D that must be met. Each vehicle must leave the depot, visit n locations, and return to the depot. M routes must be constructed based on the number of vehicles, and all locations must be visited exactly once by a vehicle. The traveled distance of vehicles between any two locations is considered symmetric and is measured in Euclidean distance. The objective of CVRP is to minimize the overall distance traveled by vehicles while ensuring locations' demand is met, and the capacity of each vehicle is not violated. The CVRP is formulated as a Mixed-Integer Linear Programming (MILP) problem to solve it to optimality with decision variables and parameters, as shown in Table 2 (Kara *et al*., 2004). The objective function (1a) is to minimize the distance traveled by the vehicles when visiting the locations. Constraints (1b) and (1c) ensure that all vehicles start their tours from the depot and return to the depot. Constraints (1d) and (1e) ensure that each location is visited exactly once. Constraint (1f) is the sub-tour elimination constraint that forbids solutions with disconnected tours. Constraint (1f), along with constraint (1g), ensures that the capacity of the vehicles is not exceeded. The CVRP can be formulated as follows:

$$\text{Min} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} D_{ij} x_{ij} \tag{1a}$$

$$\text{s.t.} \sum_{j=1}^{N} x_{0j} = \text{M} \tag{1b}$$

$$\sum_{i=1}^{N} x_{i0} = \text{M} \tag{1c}$$

$$\sum_{\substack{j=0 \\ j\neq i}}^{N} x_{ij} = 1 \ (\text{i} = 1,\ldots,\text{N}) \tag{1d}$$

$$\sum_{\substack{i=0 \\ i\neq j}}^{N} x_{ij} = 1 \ (\text{j} = 1,\ldots,\text{N}) \tag{1e}$$

$$u_i - u_j + \text{Q}x_{ij} \leq \text{Q} - D_j \ (\text{i, j} = 1,\ldots,\text{N}; \text{i} \neq \text{j}) \tag{1f}$$

$$D_i \leq u_i \leq \text{Q} \ (\text{i} = 1,\ldots,\text{N}) \tag{1g}$$

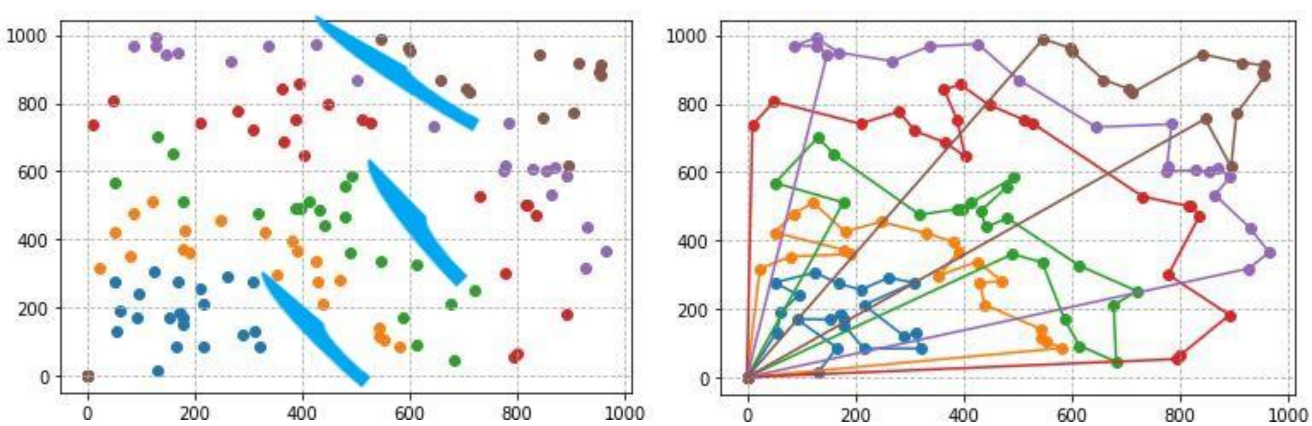
$$x_{ij} = \{0, 1\} \ \forall \ i,j \ \ (i,j = 1,..,N) \tag{1h}$$

$$u_i \geq 0 \tag{1i}$$

Table 2. Decision variables and parameters for CVRP formulation

Index Set	
i	Location where the vehicle is leaving from
j	Location where the vehicle is entering
Decision Variables	
x_{ij}	1, if there exists a route between locations *i* and *j* 0, otherwise
u_i	The cumulative demand on the route up to location *i*
Parameters	
D_{ij}	The distance traveled from location *i* to location *j*
M	The number of vehicles
Q	The capacity of vehicle
D_i	The demand of location *i*
N	The number of locations

The MILP is not easy to solve to optimality because its complexity requires long computational time and is of great concern for real-world problems (Moghaddam *et al*., 2012). Therefore, cluster-first route-second heuristics are commonly used to solve the CVRP instead of the MILP as they provide good solutions in less computational time and are considered in this research. The performance of cluster-first route-second heuristics depends on cluster formation and route construction by TSP. The focus of this research is to cluster locations while considering radial distanced patterns, as efficient clustering of locations can improve the solutions drastically. The steps of clustering and routing are illustrated in Figure 2. For the given instance, geographical barriers of rivers are present radially; therefore, in Figure 2a, the locations are clustered radially. Clustering angularly would require long detours to avoid rivers and would result in an increased traveled distance of vehicles. Radial clustering of locations when barriers are present radially can reduce the number of detours needed and, therefore, the traveled distance. In Figure 2b, the routes identified do not include detours since the clustering is performed radially and is independent of the barriers since this is tackled in the clustering phase. The clustering of such instances requires understanding locations' geography so it can be determined if radial clustering would be efficient. The algorithm is designed to determine the degree of radial and angular clustering needed depending on the nature of the locations' geography, which would result in a decreased traveled distance of vehicles. The initial weights ratio for radial and angular distances is set based on the nature of the data and how sparse it is radially and angularly; the ratio is updated to ensure that the best clustering is obtained.



(a) Clustering phase (b) Routing phase

Figure 2. Two-phase method to solve CVRP

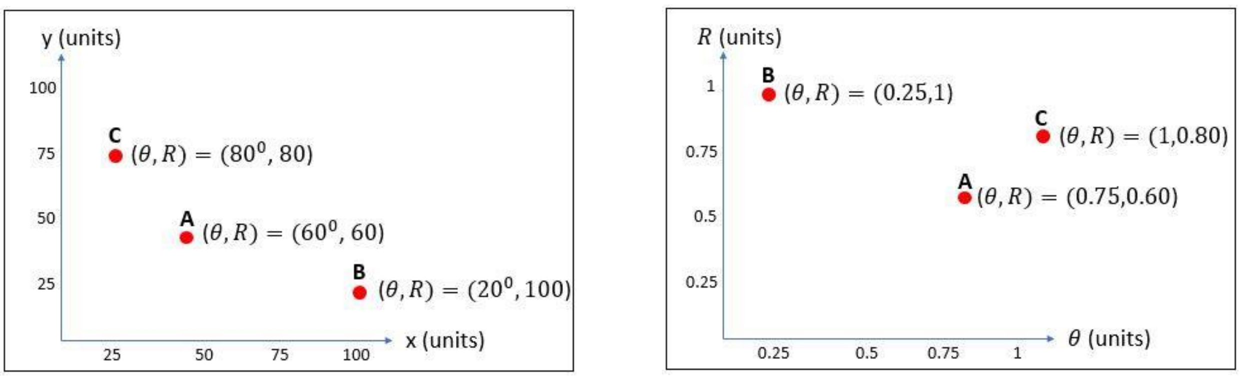
4. ANGULAR AND RADIAL DISTANCE SWEEPING ALGORITHM

For instances with radial clustered patterns, considering the radial distance of locations along with the angular distance can improve results, ensure better clustering, and minimize the traveled distance of the vehicles. Therefore, this research proposes an Angular and Radial Distanced Sweeping Algorithm (AR-SWA) that performs clustering based on locations' angular and radial distances accounting for different configurations of the distances based on locations' geography.

The steps to perform AR-SWA starts with the normalization of angular and radial distances. Clustering is then performed to the normalized locations by taking the location with the least angular distance as the first location to be visited. The nearest location is determined by calculating a new distance based on the location's angular and radial distances. The nearest location to that is also determined in the same process where locations are assigned to a vehicle until its capacity is met. The next cluster starts from the least angular distance out of the unassigned locations and follows the same steps. This process continues until all locations are assigned to vehicles. Once clusters are formed, and initial routes are constructed, each route is optimized using a 2-opt point exchange. The proposed algorithm has a polynomial time complexity of $O(n^4)$, where n represents the instance size. The detailed steps to perform AR-SWA are explained in the following sections.

4.1 Normalization of Locations' Angular and Radial Distances

The first step in AR-SWA normalizes the locations' angular and radial distances. Figure 3 shows how distances are normalized to facilitate calculations. In 3a, location B is located along a line that radiates 20 degrees above horizontal and 100 units from the origin. In 3b, which shows B's location as a proportion of the longest distance in the cluster, the ordered pair is transformed to (0.25, 1). 0.25 is the normalized angular distance based on the largest angular distance, and 1 shows it is the farthest location in the cluster radially. Because the angular and radial distances have different units, they cannot be combined or compared in their current form. This creates the need to normalize both distances to the same scale, where 0 to 1 is considered. For the radial distance, all locations' radial distances are divided by the largest radial distance of locations to get normalized radial distances. This ensures that the highest radial distance considered is 1 and that all radial distances lie between 0 and 1. For the angular distance, all locations' angular distances are divided by the largest angular distance of locations to get normalized angular distances. The detailed steps for normalization of the locations are described in Algorithm 1.



(a) Unnormalized instance (b) Normalized instance based on *0* & *R*

Figure 3. Example of normalization steps

Algorithm 1 Normalization of locations' coordinates

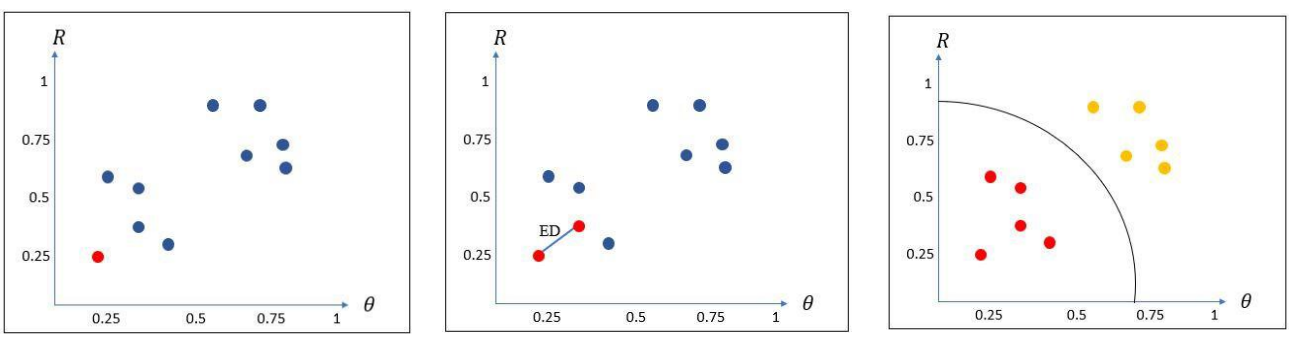
procedure IDENTIFY POLAR COORDINATES
input the number of locations N
for n in N **do**
  input location's longitude as x_n
  input location's latitude as y_n
  calculate location's angular distances $\Theta_n = tan^{-1}(y_n/x_n)$
  calculate location's radial distances $R_n = \sqrt{{x_n}^2 + {y_n}^2}$
  determine max Θ
  determine max R
for n in N **do**
  calculate normalized $\Theta_n = \Theta_n/\max \Theta_n$
  Θ_n = normalized Θ_n
  calculate normalized $R_n = R_n/\max R$
  R_n = normalized R_n

4.2 Clustering based on Nearest Location

The locations' normalized angular and radial distances identified in Algorithm 1 will be used for the clustering phase. In AR-SWA, the locations are clustered by using SWA and finding the nearest locations based on their angular and radial distances. Starting with the location with the least angular distance, as in SWA, the nearest location is determined by examining a new distance from the current location to its neighbors. The location with the least distance is the nearest neighbor. The AR-SWA calculates the new Euclidean Distance (*ED*) by considering Θ and R representing the angular and radial distances, respectively, each with a specific weight assigned of w_1 and w_2 indicating the ratio of angular and radial distances considered when calculating the *ED*. The equation used to calculate the distance between locations is as follows:

$$ED = \sqrt{\Theta^2 + (w2/w1)^2 R^2} \qquad (2)$$

Because each instance has its unique distribution of locations, some instances might be better clustered if the nearest locations are considered using a higher ratio of radial to angular distance. Therefore, the proposed heuristic starts with an initial ratio based on the geographic nature of the locations and explores other ratios that might improve the performance. Once distances are calculated, and the nearest location is identified, the process is repeated by finding the nearest location to the current location until vehicle capacity Q is met and a cluster is formed. Another cluster starts from the least angular distance out of the unassigned locations and continues by finding the nearest location by using the new distance based on *ED* and stops when vehicle capacity Q is met. This is repeated until all locations are assigned to vehicles. After the clusters are formed, an initial route for each cluster is formed based on the sequence of assigned locations to that cluster. This is illustrated in Figure 4a, where the location with the least angular distance is identified as the first location to be visited, and in 4b, the nearest location is identified by examining the angular and radial distances by calculating *ED* and choosing the location with the least *ED*. The vehicle capacity is checked after each location is assigned to a cluster; when the vehicle capacity is exceeded, the location is assigned to a second cluster. The nearest locations to that location are also assigned to the second cluster until the second vehicle capacity is met. The clusters formed through these steps are shown in 4c. The locations indicate clear radial clusters emphasizing the use of a higher ratio for radial to angular distance, which results in two clusters formed radially. The detailed steps for the clustering and initial routes assignment are described in Algorithm 2.



(a) SWA to identify location (b) Identification of nearest location (c) Clustering results with the least angular distance

Figure 4. Example of clustering using AR-SWA

Algorithm 2 Clustering and routes assignment phase

procedure CLUSTER BASED ON NEAREST LOCATION
 input the number of vehicles M
 input vehicle capacity Q
 initialize depot d as (0,0)
 for m in M **do**
  $n = 1$
  if Θ_n with respect to d is min **then**
   assign location n to m
  else
  procedure FIND MODIFIED DISTANCES BASED ON d
  initialize w_2/w_1

while n in N **do**
calculate $ED_n = \sqrt{\Theta_n^{2} + (w2/w1)^{2} R_n^{2}}$
increment w_2/w_1
calculate ED_n new
if ED_n new < ED_n **then**
$ED_n = ED_n$ new
else
$ED_n = ED_n$
repeat until ED_n new > ED_n
if ED_n is min **then**
assign location n to m
repeat until Q is met
construct initial route of n locations
else
d = min unassigned Θ_n

4.3 Vehicles Routing

After clusters are formed, and initial vehicle routes are constructed for each cluster, the routes are optimized to reduce the vehicles traveled distances with the steps shown in Algorithm 3. Many methods have been used to improve the routes obtained by solving TSP. In this research, the commonly known 2-opt point exchange is used for the routing phase. 2-opt point exchange is known for its good performance and has been widely used for solving CVRP (Lei and Li, 2010). Kaku *et al*. (2003) compared the efficiency of three exchanges for constructing routes, including swap, relocation, and 2-opt, and identified that the 2-opt is the most efficient. The route order of any two locations in the same cluster is swapped using 2-opt. If the swap results in a reduced traveled distance, then the swap is accepted, and a new route is formed with the updated route order based on the swapping. The swapping of two locations is repeated until no further swapping results in improvements which is identified as a stopping criterion. This results in routes that have minimized the traveled distance of vehicles by checking all possible swaps of locations to find the best order of locations to be visited for each vehicle.

Algorithm 3 Vehicles routing optimization phase

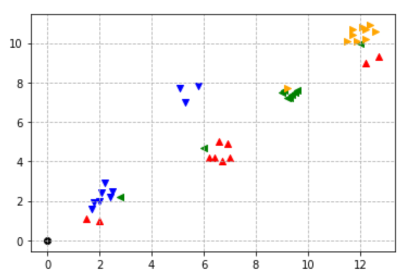
procedure OPTIMIZE VEHICLE ROUTES USING 2-OPT EXCHANGE
for n in N **do**
Θ_n = normalized Θ_n max Θ_n
R_n = normalized R_n max R_n
set $i = 1$
for m in M **do**
while n is assigned to m **do**
calculate distance from n to $n + i$
$j = i+1$
calculate new distance from n to $n + j$
if new distance < distance **then**
swap i with j
else
do not swap
$i = i+1$
repeat until new distance > distance
obtain optimized route

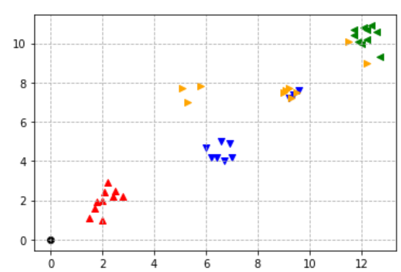
5. NUMERICAL EXPERIMENTS

Numerical experiments were conducted to evaluate the performance of AR-SWA in terms of the total distance traveled by vehicles and computational time. The experiments consist of two parts; the first part evaluates the performance of the proposed method for a targeted class of problems that are designed with radial patterns of locations and compares its performance against SWA and one of its well-known variants, SNN. The second part evaluates the performance of the proposed method on the Augerat benchmark datasets A and B (Augerat *et al*., 1995) and Christofides dataset M (Christofides and Eilon, 1969) and compares it with the SWA. All experiments were conducted using Python 3.7 on an Intel Core i7-8550U with 16 GB memory.

5.1 Performance Comparison on Dataset with Radial Patterns

Instances were generated to show locations present on radial clustered patterns and test the performance of AR-SWA for the targeted set of problems, especially since it is designed to handle locations with specific geographic nature of locations with radial patterns. For such cases, a higher ratio of radial to angular distance is considered. The performance of these instances is tested against the classical SWA and a well-known variant of SWA, SNN. The exact solution is not included in the comparison since it consumes high computational time and hence is not practical for large-scale real-world problems. Different studies omitted the comparison with exact solutions as it was not able to provide efficient solutions in reasonable computational time for large-scale problems (Ahkamiraad and Wang, 2018). Therefore, the performance comparison is tested for the heuristic methods. Figures 5 and 6 show a comparison of the clusters obtained using SWA, SNN, and AR-SWA for instances 2 and 3, each having apparent radial clusters of locations and four vehicles leaving from the depot (0,0). Clustering using AR-SWA identifies four radial clusters and results in a minimized traveled distance compared to SWA and SNN, which did not efficiently identify the radial clusters. AR-SWA considers locations' geography and groups locations that are close radially together while considering their angular distances and how sparse they are. The clusters obtained using SWA and SNN did not consider the characteristics of the locations' geography, which resulted in locations that are sparsely clustered together. The experimental results are summarized in Table 3 and show that AR-SWA heuristic finds efficient clustering for locations on radial patterns and outperforms SWA and SNN for the targeted instances in terms of the total distance traveled with improvements up to 40%. Paired t-tests were performed between AR-SWA and SWA and AR-SWA and SNN for both small-sized and large-sized instances at a 95% confidence interval to test if the results are statistically significant. The results for the tests conducted indicate a statistically significant difference between the AR-SWA and SWA and AR-SWA and SNN for small and large-sized instances with a p-value < 0.05, indicating the superiority of the proposed method.


(a) Clustering using SWA


(b) Clustering using SNN

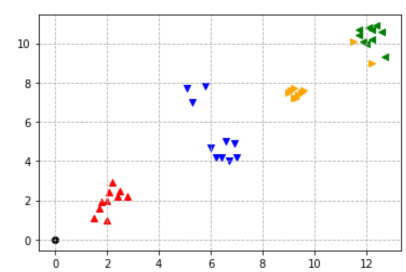
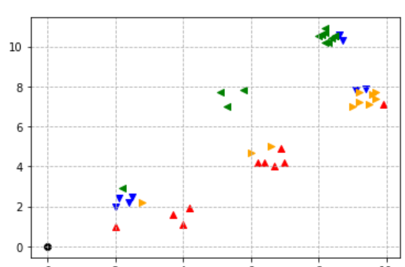
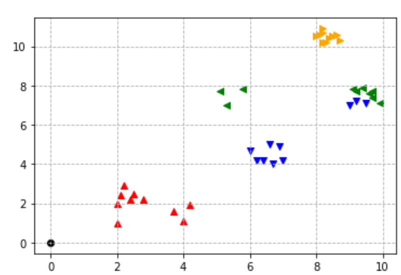

(c) Clustering using AR-SWA

Figure 5. Example of clustering using SWA, SNN, and AR-SWA, for instance, 2 with apparent radial clusters


(a) Clustering using SWA


(b) Clustering using SNN

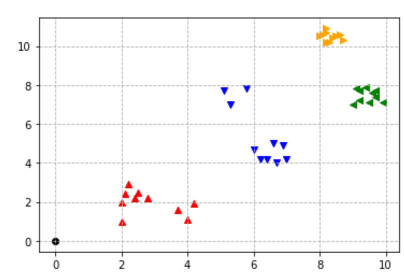

(c) Clustering using AR-SWA

Figure 6. Example of clustering using SWA, SNN, and AR-SWA, for instance, 3 with apparent radial clusters

Table 3. Comparison of total distance traveled (units) for instances with radial patterns

Size	Instance	SWA (x)	SNN (y)	AR-SWA (z)	Imp% by AR-SWA compared to SWA ((x-z)/x)	Imp% by AR-SWA compared to SNN ((y-z)/y)
Small (40-50 locations)	1	90	89	67	26	25
	2	118	111	100	15	10
	3	108	109	90	17	17
	4	76	74	55	28	26
	5	75	61	55	27	10
	6	53	55	47	11	15
	7	50	49	30	40	39
	8	44	34	34	23	0
	9	89	89	68	24	24
	10	118	111	99	16	11
	11	93	91	87	6	4
	12	86	78	74	14	6
	13	92	76	73	21	4
	14	127	115	105	17	8
	15	87	86	81	7	6
	16	103	99	95	8	5
	17	94	84	83	12	1
	18	87	83	79	9	5
	19	106	109	102	4	7
	20	102	103	92	9	10
Large (70-80 locations)	21	108	105	105	3	0
	22	112	112	110	2	2
	23	117	111	106	9	4
	24	147	133	126	14	5
	25	132	136	120	9	12
	26	183	175	170	7	3
	27	92	87	81	12	7
	28	103	101	101	2	0
	29	108	105	104	4	1
	30	167	155	139	17	11
	31	114	117	104	9	12
	32	178	175	153	14	12
	33	121	118	113	7	5
	34	145	143	128	12	11
	35	137	136	126	8	7
	36	140	136	133	5	2
	37	108	101	93	14	8
	38	102	103	95	7	8
	39	136	126	121	11	4
	40	154	158	143	7	10

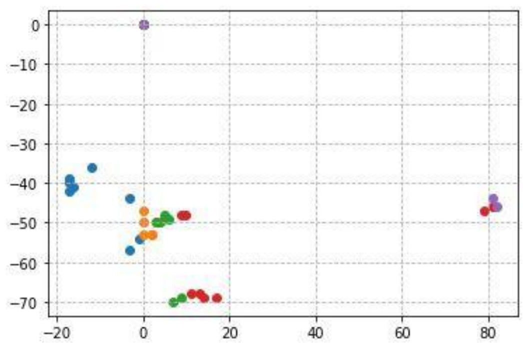
5.2 Performance Comparison on Benchmark Datasets

The proposed heuristic is tested on benchmark datasets to prove its robustness in handling different geographical characteristics of locations, although they do not necessarily have the same characteristics the heuristic is designed for. The proposed heuristic provides good solutions and outperforms SWA, resulting in a decreased traveled distance for 80% of the instances in the benchmark datasets, which do not consider radial patterns. An overall average improvement of 3% is achieved, with some instances resulting in up to 10-14% improvement compared to SWA. Table 4 summarizes the results for the benchmark datasets and shows the comparison of SWA and AR-SWA in terms of the total travel distance of vehicles. The solution of clusters using SWA and AR-SWA are analyzed and compared. Figure 8 compares the clusters formed using SWA and AR-SWA, for instance, in the benchmark dataset. 7a shows the clusters formed using SWA, where five vehicle routes are generated based on locations' angular distances. The total traveled distance of vehicles using the SWA equals 905 units. Clustering for the same instance using AR-SWA is shown in 7b, where an initial solution is obtained based on setting an initial ratio of distances based on locations' geography considering higher radial to angular distance since there are some radial clusters present. Five routes are generated based on locations' radial distances; the total distances traveled equals 861 units with an average improvement of around 5% by including the radial distance compared with SWA. This shows that considering radial distances when clustering is more efficient compared to clustering solely by angular distances, especially for instances that have some patterns of radial clusters. AR-SWA iterates until it finds the best solution by considering the geographic nature of locations. The improved solution obtained using AR-SWA is shown in 7c. The total distance equals 817 units, with an average improvement of around 10%. Since this instance has radial clusters that may not be apparent as in the generated cases, considering both angular and radial distances equally is more efficient. The best solution obtained by AR-SWA is shown in 7d with a total distance traveled of 805 units and an average improvement of around 11%. This shows that the solutions obtained using AR-SWA outperform SWA for this instance. In 7a, two vehicles must visit the right-most cluster. However, in 7d, only one vehicle is needed for the right-most cluster, indicating that the proposed method reduces the total traveled distances by accounting for radial patterns even if they are not as apparent.

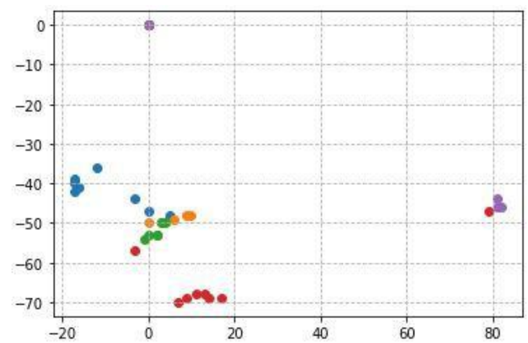
Table 4. Comparison of total distance traveled (units) and computational time (sec) for benchmark instances

Instance	SWA (x)	AR-SWA (z)	Imp% ((x-z)/x)	SWA time	AR-SWA time
A-n32-k5	990	**977**	1.24	0.0211	0.0397
A-n33-k5	910	**848**	6.77	0.0102	0.0362
A-n33-k6	1101	**1080**	1.95	0.0088	0.0184
A-n34-k5	1028	**930**	9.53	0.0121	0.0392
A-n36-k5	1064	**1039**	2.40	0.0214	0.0392
A-n37-k5	974	**972**	0.20	0.0271	0.0655
A-n37-k6	**1340**	1345	-0.37	0.0145	0.0241
A-n38-k5	1014	**925**	8.76	0.0194	0.0455
A-n39-k5	1146	**1081**	5.68	0.0214	0.0499
A-n39-k6	1223	**1181**	3.43	0.0184	0.0450
A-n44-k7	1313	**1226**	6.64	0.0221	0.0476
A-n45-k6	1252	**1217**	2.79	0.0188	0.0414
A-n45-k7	1454	**1412**	2.85	0.0243	0.0460
A-n46-k7	1180	**1126**	4.63	0.0269	0.0640
A-n48-k7	**1301**	1303	-0.20	0.0249	0.0455
A-n53-k7	1371	**1335**	2.62	0.0297	0.0473
A-n54-k7	**1544**	1561	-1.10	0.0258	0.0813
A-n55-k9	1482	**1469**	0.88	0.0188	0.0479
A-n60-k9	1919	**1916**	0.16	0.0467	0.0675
A-n61-k9	**1494**	1501	-0.49	0.0241	0.0614
A-n62-k8	1798	**1758**	2.24	0.0699	0.0936
A-n63-k10	1882	**1788**	5.0	0.0224	0.0849
A-n63-k9	2043	**1999**	2.12	0.0312	0.0699
A-n64-k9	**1832**	1896	-3.54	0.0391	0.0823
A-n65-k9	1621	**1576**	2.77	0.0304	0.0851
A-n69-k9	1539	**1504**	2.24	0.0440	0.1072
A-n80-k10	2357	**2317**	1.68	0.0514	0.1176
B-n31-k5	905	**806**	10.95	0.0127	0.0210
B-n34-k5	1282	**1158**	9.67	0.0203	0.0563
B-n35-k5	**1134**	**1134**	0.00	0.0193	0.0408
B-n38-k6	1065	**1050**	1.36	0.0187	0.0357
B-n39-k5	748	**708**	5.30	0.0244	0.0583
B-n41-k6	**1020**	1026	-0.52	0.0149	0.0450
B-n43-k6	949	**936**	1.31	0.0180	0.0575
B-n44-k7	1441	**1238**	14.10	0.0158	0.0450
B-n45-k5	1058	**1042**	1.54	0.0325	0.0743
B-n45-k6	1009	**1007**	0.21	0.0219	0.0571
B-n50-k7	**1022**	1036	-1.36	0.0364	0.0592

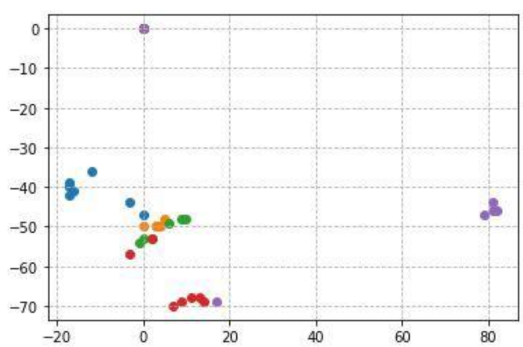
Instance	SWA (x)	AR-SWA (z)	Imp% ((x-z)/x)	SWA time	AR-SWA time
B-n50-k8	1694	**1634**	3.55	0.0544	0.0507
B-n51-k7	1471	**1420**	3.46	0.0216	0.0469
B-n52-k7	**1115**	1130	-1.39	0.0561	0.0630
B-n56-k7	1014	**959**	5.4	0.0478	0.0844
B-n57-k7	1714	**1696**	1.07	0.0314	0.0604
B-n57-k9	2009	**1863**	7.30	0.0439	0.0612
B-n63-k10	2027	**2016**	0.56	0.0567	0.0518
B-n64-k9	**1333**	1341	-0.61	0.0314	0.0906
B-n66-k9	1706	**1685**	1.24	0.0729	0.1140
B-n67-k10	1573	**1534**	2.52	0.0295	0.0793
B-n68-k9	**1648**	1698	-3.05	0.0463	0.0981
B-n78-k10	**1748**	1750	-0.12	0.0934	0.0981
M-n101-k10	1494	**1471**	1.56	0.1463	0.3095
M-n121-k7	1833	**1661**	9.40	0.5307	0.9364
M-n151-k12	1846	**1804**	2.29	0.3862	0.6598
M-n200-k16	2352	**2325**	1.14	0.4586	0.7042



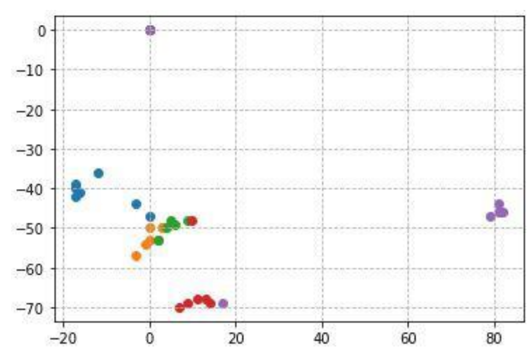
(a) B-n31-k5 clusters using SWA



(b) B-n31-k5 initial clusters using AR-SWA



(c) B-n31-k5 improved clusters using AR-SWA



(d) B-n31-k5 best clusters using AR-SWA

Figure 7. Example of clustering using SWA and AR-SWA for B-n31-k5

6. CONCLUSION

This research overcomes the limitations of the SWA for clustering instances with specific radial patterns that degrade its performance. The proposed method extends on the SWA and finds a systematic approach to solve CVRP by clustering locations based on their angular and radial distances. The proposed method outperforms SWA and results in better cluster formation and better performance in terms of the total traveled distance. AR-SWA is competitive with other methods, such as SNN, and results in better solutions for the targeted instances by reducing the traveled distance by up to 40%. The proposed method was tested on 53 instances of benchmark datasets. Almost 80% of instances resulted in a decreased traveled distance, where the average improvement was around 3%. AR-SWA is simple to implement and results in better solutions for most instances, with an insignificant increase in the computational time needed compared with SWA. This method is suitable for real-world applications with apparent radial clusters and distinct geographical layouts of locations.

REFERENCES

Ahkamiraad, A. and Wang, Y. (2018). Capacitated and Multiple Cross-Docked Vehicle Routing Problem with Pickup, Delivery, and Time Windows. *Computers & Industrial Engineering*, 119:76–84.

Akhand, M., Peya, Z. J., and Murase, K. (2017). Capacitated Vehicle Routing Problem Solving Using Adaptive Sweep and Velocity Tentative Pso. *International Journal of Advanced Computer Science and Applications*, 8(12): 288–295.

Altınel, İ. K. and Öncan, T. (2005). A New Enhancement of The Clarke and Wright Savings Heuristic for The Capacitated Vehicle Routing Problem. *Journal of The Operational Research Society*, 56(8): 954–961.

Anbuudayasankar, S., Ganesh, K., Koh, S. L., and Ducq, Y. (2012). Modified Savings Heuristics and Genetic Algorithm for Bi-Objective Vehicle Routing Problem with Forced Backhauls. *Expert Systems with Applications*, 39(3): 2296–2305.

Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., and Rinaldi, G. (1995). Computational Results with A Branch and Cut Code for The Capacitated Vehicle Routing Problem. *Institut National Polytechnique (France)*.

Baldacci, R., Hadjiconstantinou, E., and Mingozzi, A. (2004). An Exact Algorithm for The Capacitated Vehicle Routing Problem Based on A Two-Commodity Network Flow Formulation. *Operations Research*, 52(5): 723–738. Beasley, J. E. (1983). Route First Cluster Second Methods for Vehicle Routing. *Omega*, 11(4): 403–408.

Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I. (2016). The Vehicle Routing Problem: State of The Art Classification and Review. *Computers & Industrial Engineering*, 99:300–313.

Chen, M. H., Chang, P. C., Chiu, C. Y., and Annadurai, S. P. (2015). A Hybrid Two-Stage Sweep Algorithm for Capacitated Vehicle Routing Problem. in *2015 International Conference on Control, Automation, and Robotics*, Pages 195–199. IEEE.

Chen, X. and Zhao, J. (2013). Bidding to Drive: Car License Auction Policy in Shanghai and Its Public Acceptance. *Transport Policy*, 27:39–52.

Christofides, N. and Eilon, S. (1969). An Algorithm for The Vehicle-Dispatching Problem. *Journal of The Operational Research Society*, 20(3): 309–318.

Cinar, D., Gakis, K., and Pardalos, P. M. (2016). A 2-Phase Constructive Algorithm for Cumulative Vehicle Routing Problems with Limited Duration. *Expert Systems with Applications*, 56:48–58.

Clarke, G. and Wright, J. W. (1964). Scheduling of Vehicles from A Central Depot to A Number of Delivery Points. *Operations Research*, 12(4): 568–581.

Comert, S. E. and Yazgan, H. R. (2021). Effective Cluster-First Route-Second Approaches Using Metaheuristic Algorithms for The Capacitated Vehicle Routing Problem. *International Journal of Industrial Engineering*, *28*(1).

Fukasawa, R., Longo, H., Lysgaard, J., De Aragão, M. P., Reis, M., Uchoa, E., and Werneck, R. F. (2006). Robust Branch-And-Cut-And-Price for The Capacitated Vehicle Routing Problem. *Mathematical Programming*, 106(3): 491–511.

Gillett, B. E. and Miller, L. R. (1974). A Heuristic Algorithm for The Vehicle-Dispatch Problem. *Operations Research*, 22(2): 340–349.

Kaku, I., Xiao, Y., and Xia, G. (2003). The Deterministic Annealing Algorithms for Vehicle Routing Problems. *International Journal of Smart Engineering System Design*, 5(4): 327–339.

Kalatzantonakis, P., Sifaleras, A., and Samaras, N. (2019). Cooperative Versus Non-Cooperative Parallel Variable Neighborhood Search Strategies: A Case Study on The Capacitated Vehicle Routing Problem. *Journal of Global Optimization*, Pages 1–22.

Kara, I., Laporte, G., and Bektas, T. (2004). A Note on The Lifted Miller–Tucker–Zemlin Subtour Elimination Constraints for The Capacitated Vehicle Routing Problem. *European Journal of Operational Research*, 158(3): 793–795.

Lei, J. J. and Li, J. (2010). Solving Capacitated Vehicle Routing Problems by Modified Differential Evolution. in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, Volume 2, Pages 513–516. IEEE.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A New Branch-And-Cut Algorithm for The Capacitated Vehicle Routing Problem. *Mathematical Programming*, 100(2): 423–445.

Marinelli, M., Colovic, A., and Dell'Orco, M. (2018). A Novel Dynamic Programming Approach for Two-Echelon Capacitated Vehicle Routing Problem in City Logistics with Environmental Considerations. *Transportation Research Procedia*, 30:147–156.

Moghaddam, B. F., Ruiz, R., and Sadjadi, S. J. (2012). Vehicle Routing Problem with Uncertain Demands: An Advanced Particle Swarm Algorithm. *Computers & Industrial Engineering*, 62(1): 306–317.

Na, B., Jun, Y., and Kim, B. I. (2011). Some Extensions to The Sweep Algorithm. *The International Journal of Advanced Manufacturing Technology*, 56(9): 1057–1067.

Peya, Z. J., Akhand, M., and Murase, K. (2018). Capacitated Vehicle Routing Problem Solving Through Adaptive Sweep Based Clustering Plus Swarm Intelligence Based Route Optimization. *Oriental Journal of Computer Science and Technology*, 11(2): 88–102.

Peya, Z. J., Akhand, M., Sultana, T., and Rahman, M. H. (2019). Distance Based Sweep Nearest Algorithm to Solve Capacitated Vehicle Routing Problem. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10(10).
Pichpibul, T. and Kawtummachai, R. (2012). An Improved Clarke and Wright Savings Algorithm for The Capacitated Vehicle Routing Problem. *Scienceasia*, 38(3): 307–318.

Rattanamanee, T., Nanthavanij, S. and Dumrongsiri, A. (2020). Heuristic Procedure for Multi-Workday Vehicle Routing Problem with Pre-Assigned Workers and Manual Unloading. *International Journal of Industrial Engineering*, 27(3).

Renaud, J. and Boctor, F. F. (2002). A Sweep-Based Algorithm for The Fleet Size and Mix Vehicle Routing Problem. *European Journal of Operational Research*, 140(3): 618–628.

Savitri, H. and Kurniawati, D. (2018). Sweep Algorithm and Mixed Integer Linear Program for Vehicle Routing Problem with Time Windows. *Journal of Advanced Manufacturing Systems*, 17(04): 505–513.

Suthikarnnarunai, N. (2008). A Sweep Algorithm for The Mix Fleet Vehicle Routing Problem. *in Proceedings of The International Multi Conference of Engineers and Computer Scientists*, Volume 2, Pages 19–21. Citeseer.

Tarawneh, L., Alelaumi, S., and Yoon, S. W. (2020). Modified Sweeping Approach for Solving Capacitated Vehicle Routing Problem. in *IIE Annual Conference*. *Proceedings* (Pp. 1-6). Institute of Industrial and Systems Engineers (IISE).

Teodorović, D. and Pavković, G. (1996). The Fuzzy Set Theory Approach to The Vehicle Routing Problem when Demand at Nodes is Uncertain. *Fuzzy Sets and Systems*, 82(3): 307–317.

Yücenur, G. N. and Demirel, N. C. (2011). A New Geometric Shape-Based Genetic Clustering Algorithm for The Multi-Depot Vehicle Routing Problem. *Expert Systems with Applications*, 38(9): 11859–11865.

Zahrul, J. P., Akhand, M. A. H., Sultana, T., and Rahman, M. H. (2019). Distance Based Sweep Nearest Algorithm to Solve Capacitated Vehicle Routing Problem. *International Journal of Advanced Computer Science and Applications*, *10*(10).

Zajac, S. (2018). on A Two-Phase Solution Approach for The Bi-Objective K-Dissimilar Vehicle Routing Problem. *Journal of Heuristics*, 24(3): 515–550.

Zhang, S., Lee, C., Choy, K., Ho, W., and Ip, W. (2014). Design and Development of A Hybrid Artificial Bee Colony Algorithm for The Environmental Vehicle Routing Problem. *Transportation Research Part D: Transport and Environment*, 31:85-99.