Jurnal
Teknologi

Full Paper

ANT COLONY OPTIMIZATION ALGORITHM WITH SEQUENTIAL VARIABLE NEIGHBOURHOOD SEARCH CHANGE STEP IN THE WASTE COLLECTION SYSTEM

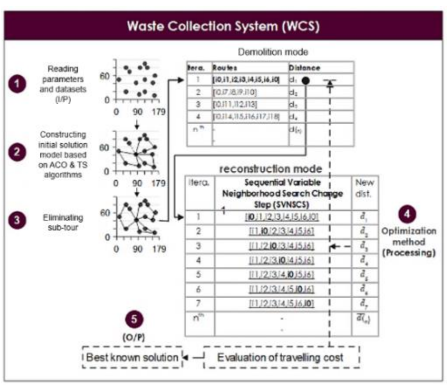
Thaeer Mueen Sahib[a,b], Rosmiwati Mohd-Mokhtar[a]*, Azleena Mohd-Kassim[c]

[a]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia
[b]Department of Electrical Techniques, Kufa Technical Institute, Al-Furat Al-Awsat Technical University, 54003, Iraq
[c]School of Computer Science, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang, Malaysia

Article history
Received
4 January 2023
Received in revised form
20 November 2024
Accepted
3 December 2024
Published Online
26 June 2025

*Corresponding author
eerosmiwati@usm.my

Graphical abstract



Abstract

This study improves the exploration of ant colony optimization (ACO) ability by adapting it with the Sequential Variable Neighbourhood Search Change Step (SVNSCS) algorithm as post-improvements for solving the Capacitated Vehicle Routing Problem (CVRP) in solid waste collection methodology. The aim is to reduce the cost of waste collection by minimizing the route distance and the number of vehicles to serve all containers within the route. Technically, SVNSCS explores the search space depending on the information associated with the demands and coordinates of the nodes. Based on the result, the proposed algorithm demonstrates its superiority over the traditional ACO algorithm by having 66.7%, 81.81%, 62.5%, and 77.77% in terms of the best solution for four CVRP benchmark datasets of A, B, E, and P, respectively. Each dataset has different characteristics, such as the number of containers, vehicle capacity, objective function, weight of each container, and geographical distribution.

Keywords: Ant colony optimization, capacitated vehicle routing problem, optimization algorithm, sequential variable neighbourhood search change step, solid waste collection

Abstrak

Kajian ini menambah baik penerokaan keupayaan pengoptimuman koloni semut (ACO) dengan mengadaptasinya dengan algoritma langkah perubahan carian kejiranan pembolehubah berurutan (SVNSCS) sebagai penambahbaikan pasca dalam menyelesaikan masalah penghalaan kenderaan berkapasiti (CVRP) dalam metodologi pengumpulan sisa pepejal. Tujuannya untuk mengurangkan kos kutipan sisa dengan meminimumkan jarak laluan dan bilangan kenderaan untuk melayani semua kontena dalam laluan. Secara teknikal, SVNSCS meneroka ruang carian bergantung kepada maklumat yang berkaitan dengan permintaan dan koordinat nod. Berdasarkan keputusan, algoritma yang dicadangkan mempamerkan keunggulannya berbanding dengan algoritma ACO tradisional dengan mempunyai 66.7%, 81.81%, 62.5%, dan 77.77% dari segi penyelesaian terbaik untuk empat set data penanda aras CVRP masing-masing A, B, E dan P. Setiap set data mempunyai ciri yang berbeza dari segi bilangan bekas, kapasiti kenderaan, fungsi objektif, berat setiap bekas dan agihan geografi.

Kata kunci: Pengoptimuman koloni semut, masalah penghalaan kenderaan berkapasiti, algoritma pengoptimuman, langkah perubahan carian kejiranan pembolehubah berurutan, pengumpulan sisa pepejal

1.0 INTRODUCTION

The increasing population growth, urbanization, and economic development lead to an increase in the amount of waste, while sustainable management still represents a major challenge worldwide [1, 2]. This issue led to an increase in the generation of Municipal Solid Waste (MSW) from 1.6 to 3.4 billion tons until the year 2050 globally [3]. East Asia and the Pacific had the highest share and amounted to about 23.0% [4]. Thus, local authorities face big challenges in MSW management in most developing countries [5, 6, 7].

Health risk is one of the challenges arising from the accumulation of waste, affecting public health [8]. Furthermore, it is known that failure to treat waste early at collection points will lead to the deterioration of the surrounding area and cause social issues [9, 10]. In addition, 19% of waste ends up in our drains, which then causes flash floods and drainage blockage, influencing the sustainability of human life, based on a study conducted in Kuala Lumpur, Malaysia [11]. These issues may happen because most municipalities still use a traditional way to collect solid waste [12]. Therefore, waste collection, among other stages of municipal waste management (MWM), is one of the significant strategies for minimizing total costs because it consumes 60% - 80% of the budgets of other MWM [13]. According to that, many strategies are utilized to tackle waste collection problems, such as spatial treatment controllers, as in the recent simulation study proposed by Sahib *et al.* [14]. The development of vehicle routing systems reduces total transportation distance and increases waste collection efficiency [15, 16].

However, most routing problems in waste collection are divided into two essential problems: the arc routing problem that deals with the waste along the street and the node routing problem that considers the nodes' locations as their priority [17]. Hence, there is a study concerning the use of the Geographic Information System (GIS) to improve waste collection [18]. Note that the classic CVRP involves starting the vehicles to serve all waste in a certain network and unloading that waste in the same location (depot) for disposal or transferring it to the vehicle with a bigger capacity, as shown in Figure 1.

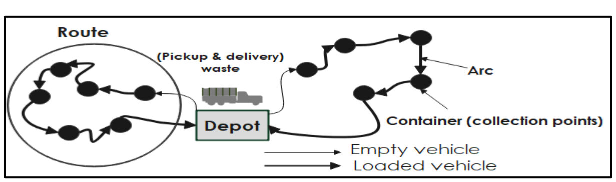


Figure 1 Pickup and delivery of waste in the same location

In this paper, we will focus on solving the homogenous Capacitated Vehicle Routing Problem (CVRP) in the application of waste collection using a metaheuristic optimization algorithm for the pickup and delivery of waste in optimal routes. Thus, the remainder of this paper is organized as follows. Section 2 represents related work, and Section 3 represents the methodology. Section 4 presents the results and discussion of the findings. Finally, Section 5 gives a conclusion and future direction.

2.0 RELATED WORK

In recent years, optimization algorithms have been proposed to solve the Vehicle Route Problem (VRP) [19, 20]. Using the optimization algorithm in the exploration stage, we can examine neighbours in the solution iteratively [21, 22]. This section will discuss the improvement of metaheuristic methods with local search heuristic algorithms for solving CVRP in a waste collection system. Regarding a recent survey, we focused on metaheuristic methods for solving routing problems in waste collection, where two studies utilized a Simulated Annealing (SA) with a Heuristic Algorithm (HA). Consequently, another study employed the Ant Colony Optimization (ACO) algorithm. However, most methods considered the Capacitated Arc Routing Problem (CARP), except one study utilized ACO to minimize the travelling time of the slowest vehicle, hence avoiding the vehicle's accident in the disposal station. Note that the latter applied only four nodes in the improvement phase, which is insufficient to evaluate the proposed system [23, 24, 25, 26].

Other studies improved their performance based on the local search strategy. One of these studies proposed Iterated Greedy (IG) with Variable Neighbourhood Search (VNS) to explore three different changes in the structure of the neighbourhoods. The list of changes is as follows: insert, exchange, and 2-opt under deconstruction and reconstruction solution. The authors also presented the greedy-random among variants to obtain a high-quality solution in the diversity stage. Although the paper is sensitive to the performance of the proposed algorithm under the intensification stage, the study did not compare the outcome with the Best-Known Solution (BKS) and other methods. In addition, it considered self-generated data only [27].

On the contrary, a study compared the proposed algorithm with BKS with other optimization methods using different dataset classes from the best-known datasets. The authors modified the Particle Swarm Optimization (PSO) algorithm with local improvement using four methods called 2-opt*, Or-opt-1, 2-opt, and Or-opt, where these changes are selected randomly (next neighbourhood unknown). The aim is to determine the best waste collection and route optimization solutions. This study considered real-time for collecting the waste from Collection Points (CPs) under a simulation environment. It was neither focused on the efficiency of the proposed algorithm opposite each iteration nor the distribution of weight in each route [9]. The weights on the edges may also represent the distance between two points and the travel time [28].

A similar study utilized a modified Backtracking Search Algorithm (BSA) with the same local improvement methods [29]. A hybrid metaheuristic algorithm also considered the SA algorithm and improved the solution by employing the VNS algorithm. Note that the VNS used local methods, including insert and removal methods, and the objective was to select the appropriate bins for the best route. This study generates real data randomly to evaluate the entire system since the next neighbourhood procedure depends on improvements. In other words, if the runtime is less or equal to 2 hours, the algorithm will move to the next procedure or stop [30].

A study in [31] proposed a population-based hybrid metaheuristic approach consisting of a Large Neighbourhood Search (LNS) and various improvement algorithms to solve the Rollon–Rolloff vehicle routing problem. The objective is to determine routes that minimize the number of required tractors and the minimum amount of total nonproductive (deadhead) time to serve all given customer demands. A modified version of the sweep nearest algorithm was proposed to overcome the weaknesses of grouping stops. The generated solutions are improved by four local search improvement algorithms, including four times of m-n exchange improvement (inter-route), crossover exchange improvement (inter-route), route reduction improvement (inter-route), and 2-opt edge exchange improvement (intra-route).

A hybrid ant colony-variable neighbourhood search algorithm was used in [32] to solve multi-depot green vehicle routing problems. Three vehicle routing models were developed focusing on multi-depot, emission, and total cost reduction. The ACO variable neighbourhood search (ACO-VNS) algorithm was tested on randomly generated small and large-scale instances, and the comparison was made with standard ACO. An improvement of 8% was observed in total cost when a comparison was made between the two algorithms on large-scale data.

However, from the authors' knowledge, there is no study focused on adapting an ACO with a VNS algorithm in terms of the Sequential Variable Neighbourhood Search Change Step (SVNSCS) for solving the CVRP, specifically addressing the issues of selecting the first node randomly in each route that belongs to the individual solution. It is considered a critical issue that must be highlighted, in addition to eliminating the sub-tour constructed (a route with a single node).

The aims of our paper will fill this research gap to improve its exploration ability and examine the performance of the basic ACO after improving the efficiency of the waste collection system. In addition, the result of the proposed algorithm is compared against the basic ACO algorithm and Best-Known Solution (BKS). The concept of contribution parts is inspired by the original paper organized by Hansen *et al.* [33], highlighted in the hierarchical scheme shown in Figure 2.

This study summarized the main contributions as follows: (1) adapting the standard ACO algorithm for solving Capacitated Vehicle Routing Problem (CVRP) in the solid waste collection system, in which the amount of waste per container and the tightness (percentage of waste amounts in the vehicle) have been calculated to follow up the profit of waste collected by vehicles, (2) designing a sub-tour elimination technique that may be generated in the solution thus reducing waste collection cost, (3) improvement of the best solution of ACO by post-optimization using SVNSCS algorithm. The waste collection system will choose the best route, in which its total amount of waste is set to be < 85% of the rest tightness of the vehicle to narrow the search space and speed up convergence, as well as the application of the SVNSCS algorithm to improve the cost of the route in terms of reducing route distance.

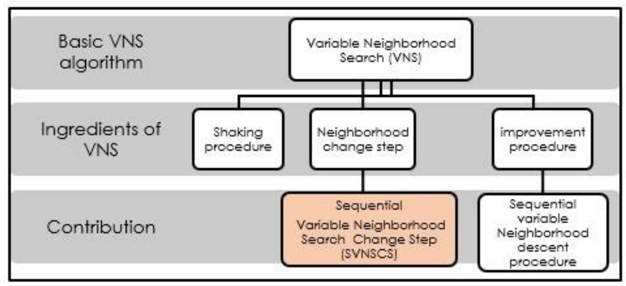


Figure 2 Ingredients of variable neighbourhood search

3.0 METHODOLOGY

3.1 Mathematical Model

The CVRP problem can be illustrated schematically, as shown in Figure 3. The CVRP distribution, $G = (N, E)$ where N represents a set of nodes graphically distributed, $N = \{C_0, C_1, \cdots, C_n\}$. The index of the depot is 0, while the other index represents the containers, which are from 1 to n. E is a set of edges called $V = \{E_0, E_1, \cdots, E_n\}$ in which each edge $(i,j) \in E = \{(i,j): i,j \in V, i \neq j\}$ and it has a positive trip cost named C_{ij} [34].

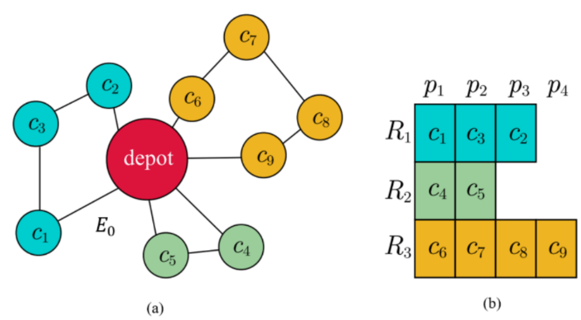


Figure 3 Representation of CVRP based on (a) graphical solution with $N = 9$ customers distributed over $k = 3$ routes, and (b) individual matrix [35]

From a practical perspective, the weights on the edges represent the travel distance from container (*i*) to container (*j*), where $(\boldsymbol{i} \neq \boldsymbol{j})$. Each container $\boldsymbol{i} \in \boldsymbol{V}' = \boldsymbol{V} \backslash \{\boldsymbol{0}\}$ has a demand $\boldsymbol{q_i}$ where $\{\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{n}\}$. For a single depot 0, having a set of homogeneous vehicles *K* with capacity limit Q, we assume the number of vehicles inside the depot is not specified. Moreover, a sub-tour elimination technique has been designed to avoid the high cost of adapting the basic ACO in classical CVRP. The sub-tour means that one vehicle serves only one or two containers in a single route. However, to formulate the waste collection system, first, we must know that the whole system is subject to the decision variable $(\boldsymbol{X_{ijk}})$. $\boldsymbol{X_{ijk}} = \boldsymbol{1}$ if vehicle k travels from a container $(\boldsymbol{i})$ to container $(\boldsymbol{j})$ under the threshold control (TC) component; otherwise, $\boldsymbol{X_{ijk}} = \boldsymbol{0}$. Thus, the vehicle does not travel on the planned route, as shown in Equation (1).

$$\boldsymbol{X_{ijk}} = \begin{cases} \mathbf{1}, \text{ if vehicle served the edge } (\boldsymbol{i},\boldsymbol{j}) \text{ under condition TC component} \\ \mathbf{0}, \quad \text{Otherwise} \end{cases} \tag{1}$$

Where

TC component = AWR ≤ 85% out of C^k

AWR = Total amount of waste in the assigned route

C^k = Vehicle Capacity.

The objective function and the constraints of the problem have been formulated as follows.

$$\text{Minimize } \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{k} d_{ij}\, X_{ijk} \tag{2}$$

$$d_{ij} = \begin{bmatrix} 0 & d_{12} & \ldots & d_{1N} \\ d_{12} & 0 & \ldots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1N} & d_{2N} & \ldots & d_{NN} \end{bmatrix} \tag{3}$$

Subject to:

$$\sum_{j=1}^{N}\sum_{k=1}^{k} X_{0jk} = 1 \tag{4}$$

$$\sum_{j=1}^{N} q_{0jk} = 0, k \in \{1,\ 2 \ldots K\} \tag{5}$$

$$\sum_{k=1}^{k}\sum_{j=0}^{N} X_{ijk} = 1, i \in \{1,2 \ldots N\}: i \neq j \tag{6}$$

$$\sum_{k=1}^{k}\sum_{i=0}^{N} X_{ijk} = 1, j \in \{1,2 \ldots N\}: i \neq j \tag{7}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} Q_j\, X_{ijk} \leq Q\ , k \in \{1,2,\ldots K\} \tag{8}$$

$$\sum_{j=1}^{N} X_{ijk} = \sum_{j=1}^{N} X_{jik} = Y_{ik}, i \in \{1,2,\ldots N\}, k \in \{1,2,\ldots K \tag{9}$$

$$\sum_{i=0}^{N}\sum_{k=1}^{K} q_{jik} - \sum_{i=0}^{N}\sum_{k=1}^{K} q_{ijk} = c_j, j \in \{1,2,\ldots n\} \tag{10}$$

$$\sum_{j=1}^{N} X_{ij}^{k} = \sum_{j=1}^{N} X_{ji}^{k} \leq 1, \text{for } i = 0 \text{ and } k \in \{1,2,\ldots k\} \text{ for } i = 0 \text{ and } k \in \{1,2,\ldots k\} \tag{11}$$

$$dist_{ij} = dist_{ji} \tag{12}$$

$$\sum_{j=1}^{N} X_{0j}^{k} \leq 1, k \in \{1,2,\ldots k \tag{13}$$

$$\sum_{i=1}^{N} X_{i0}^{k} \leq 1, k \in \{1,2,\ldots k\} \tag{14}$$

The objective function is to minimize the total distance vehicles travel to serve all containers in the network, as shown in Equation (2). Here, d_{ij} is a symmetrical matrix or a square matrix of size (*N* × *N*), then $(d_{ij} = d_{ji})$, to determine the cost of distance whether the vehicle has travelled from container (*i*) to container (*j*) based on the logical component (X_{ijk}) as Equation (3).

It should be noted that strategic systems such as waste collection systems are often subject to constraints. In this study, vehicle *K* will start its route from the depot (0) according to Equation (4). Each vehicle should start its work empty, as Equation (5) indicates. Each container can pick up the waste from it by one vehicle, as shown in Equation (6) and Equation (7). Nevertheless, Equation (8) ensures that the total amount of waste in each container cannot exceed the vehicle capacity Q. Equation (9) ensures the continuity condition. Equation (10) ensures that the vehicle empties the container visited. Equation (11) guarantees that each vehicle tour starts from the depot and ends at the same depot. Equation (12) ensures the distance between pair containers is the same. Equation (13) and Equation (14) ensure each vehicle can be used once or not be used.

3.2 Tightness Model

The tightness factors T_j^k is a significant variable depending on the profit of waste quantity in the route P_j, which is a maximize the sum of product and the quantity of waste in each container q_j in the route with the decision variable X_{ijk} in case the waste amount is a non-negative value based on Equation (15).

$$P_{jk} = max \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{k} q_j\, .X_{ijk} > 0, j \in \{2, \ldots N\} \tag{15}$$

Where: N: Number of containers in the route.

The tightness factors in the route (T_j^k) can be determined by dividing the profit value (P_j) over the vehicle capacity (C^k), as shown in Equation (16).

$$T_j^k = \left(P_j / C^k\right) > 0, j \in \{2, \ldots N\}, k \in K \tag{16}$$

The average tightness for all routes (T_{avg}^K) in the improved solution can be calculated based on Equation (17).

$$T_{avg}^{K} = \left(\sum \frac{T_j^k}{count\ of\ T_j^k}\right),\ j \in \{2, \ldots N\}, k \in K \tag{17}$$

A study on the capacity constraint of the waste-lifting vehicle results in two types of benefits. The first is related to the performance efficiency aspect of the proposed algorithm in increasing the convergence by narrowing the search space and thus reducing the execution time. Second, it relates to the technical aspect of the service vehicles used to measure the impact of high waste loading in the vehicle against fuel consumption. As a result, the solid waste collection system consists of a pre-processing module used for coding the text document of the benchmark dataset in instructions known as “regular expression operations.” Apart from that, it is configured to be able to read by Python code as well as a processing module that manages reading the dataset and parameters of the proposed algorithm and then processing the data inside it. The next section will focus on the processing module and all models employed according to the framework of the ACO algorithm, with the proposed algorithm shown in Figure 4.

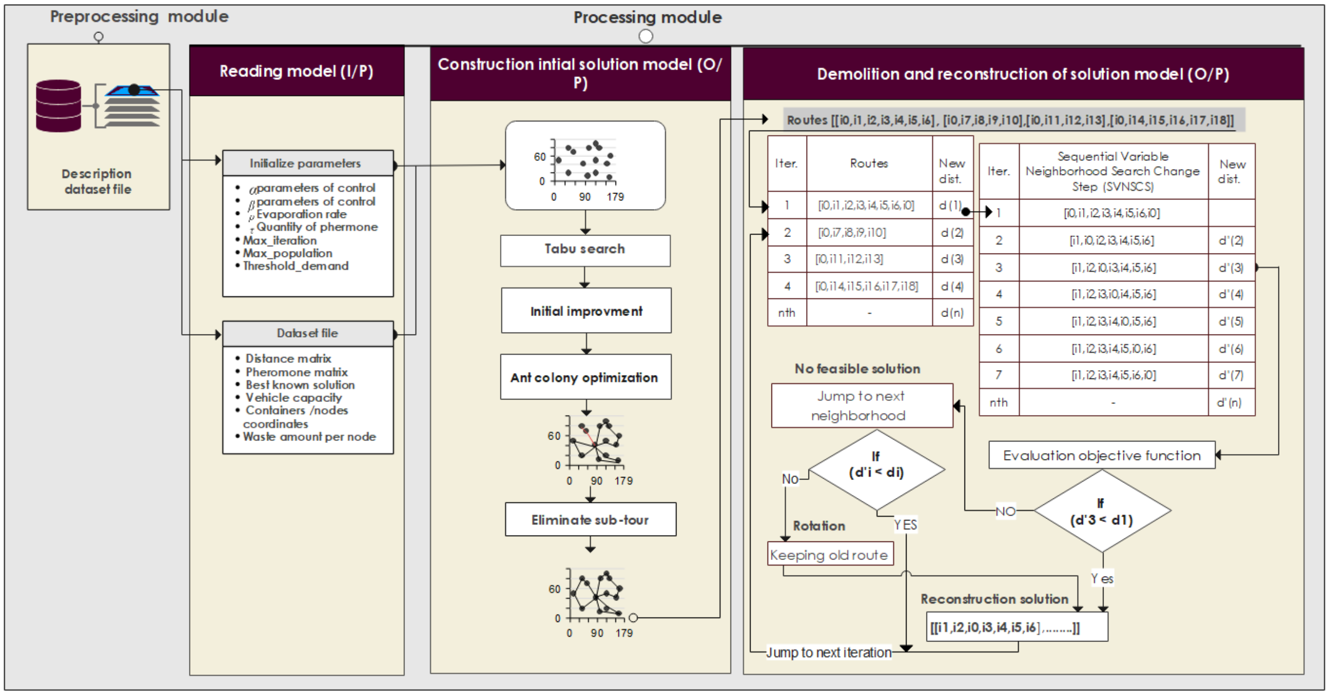


Figure 4 Framework of the proposed algorithm for reconstructing the route of waste collecting vehicle

3.3 Development of ACO

ACO is a nature-inspired and population-based metaheuristic, which was first proposed by Dorigo in 1992 and was modified in 2002 by Parpinelli *et al*. [36].

Typically, ACO is a swarm intelligence method inspired by the social behaviour of ant colonies for solving optimization problems [32, 37]. When ants are finding the food and return to the nest, they will follow the shortest path. Naturally, ants deposit pheromones on the ground during their journey to identify the routes other colony members should follow. The high quantity of the pheromones dropped by ants represents the inverse of the longest distance by vehicle during the waste collection. Although ACO is robust and can explore a solution, the convergence is still slow and down into the local optimum [38]. Therefore, improvement is required. As a result, all steps of the development and improvement of ACO in the next subsection will be structured, as shown in Figure 5.

3.3.1 Initialization of Parameters

Consider the first step of a reading model in the processing module. This section must set the number of ants, maximum iteration number, parameter control α and $\boldsymbol{\beta}$ as well as the threshold of demands for the sensitive response of the proposed algorithm in the exploitation stage in addition to the evaporation value that is set among duration [0,1] as shown in step 1.

3.3.2 Tabu Search Algorithm

This section explains the concept of the Tabu Search (TS) technique and reinforcement with some mathematical models. In contrast, the TS is a metaheuristic method introduced by Glover in 1990 that represents one of the local search heuristic algorithms capable of escaping local optima through a Tabu list and neighbourhood generation [39].

The Tabu list technique is important in clustering the routes and removing the containers not visited in the candidate containers list. This technique prevents the vehicle from visiting any container again, based on Equation 8. Instead, the search will choose a certain container and stop according to the amount of waste collected without exceeding vehicle capacity, as shown in Equation 6.

This process is done when selecting a certain container and assuming its demand exceeds vehicle capacity (violation of the constraint capacity). The TS will also remove that container to the next route while keeping the obtained route inside the main path. After the TS starts again to choose randomly another first container, this technique is iterated until other containers are chosen, as in Step 2 in Figure 5.

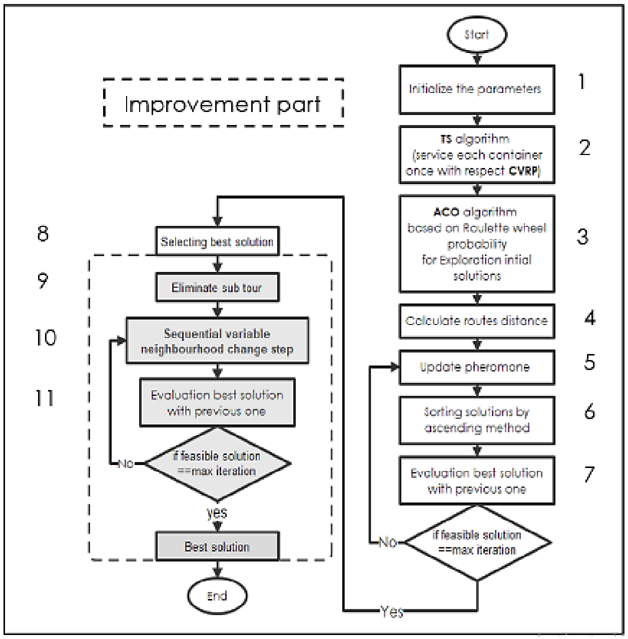


Figure 5 Flowchart of adapting ACO with SVNSCS algorithm

3.3.3 Exploration Initial Solution

Typically, the initial solution to the routing problem consists of a set of routes with a fixed limited number of containers. Furthermore, each solution is constructed based on the TS technique with assisting a probability distribution, as shown in Step 3. Note that all containers are selected relative to the first randomly selected container. This technique adopted a roulette wheel probability equation as in Equation (18). The highest probability $(P_{i,j}^k)$ of selecting the next container is the closest waste container.

$$P_{i,j}^k = \begin{cases} \frac{(\Im_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{i\in N_i^k} (\Im_{ij})^\alpha (\eta_{ij})^\beta}, & \text{if } j \in N_i^k \\ 0 & \text{,otherwise} \end{cases} \quad (18)$$

Where the term $\Im_{ij}$ = pheromone intensity in edges (*i*, *j*). The term $N_i^k =$ the feasible neighbourhood of ant *k* at the container (*i*). The η_{ij} refers to the distance between the containers (*i,j*). Meanwhile, α and β both represent the effect of controlling the pheromone intensity allocated on edge (*i, j*) and the desirability of edge (*i, j*), respectively [40]. The shortest path distance in edges (*i,j*) can be calculated based on the visibility value η_{ij} in Equation (19) [41].

$$\eta_{ij} = 1/\left(D_{(i,j)}\right) \quad (19)$$

$D_{(i,j)}$: Distance between C_i and C_j

3.3.4 Calculate Routes Distance

The total distance in each route depends on the distance matrix (symmetric matrix) to side the initial pheromone matrix, which represents the most important variables for exploring the best route. Here, the distance between container (*i*) and container (*j*) is calculated using Equation 20.

$$\text{Edges } (_{i,j}) = d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (20)$$

Where
x: Location of container source.
y: Location of container target

In Algorithm 1, two (for loops), as shown in Step 4 and Step 6, are known as the indexes of each route individually and indexes of all routes in the solution, respectively. In Step 5, node 1 refers to the depot, while Step 8 is a procedure of summing the distances from the depot to the last item in the sub-list. Finally, Step 9 is the second formula to calculate the distance between the last item in the sub-list (individual route) and the depot.

Algorithm 1 Pseudocode of calculating distance

```
1   Input: solution, edges, demands, Vehicle_Capacity
2   Output: distance for one solution
3   Procedure CalculateDistance (solution, edges,
        demand, Vehicle_Capacity)
4     for i := solution do
5        a ← 1  # refer to the depot.
6       for j := i do
7           b ←j
8           su ←su+ edges[(min(a,b), max(a,b))]
9           s_p1←(edges[(min(a,b), max(a,b))])
10          a ← b
11         b←1
12      End for
13    End for
14   return su
```

In Algorithm 2, we calculate each route's summation of demands (waste amount). Therefore, only one (for loop) is necessary to determine the vehicle capacity's tightness.

Algorithm 2 Pseudocode of calculating tightness

```
Input: edges, demands, Vehicle_Capacity
Output: tightness and amount of waste.
for j := i do
    Demands←(demand[j])
    wasteCollection←sum (Demands)
    T←wasteCollection/Vehicle_Capacity
    print "Route Tightness," T
    t←T
End for
```

3.3.5 Update of Pheromones

We create solutions for each iteration based on a pheromones update, where the evaporation and intensification procedures are based on the information from high-goodness solutions. The pheromone update aims to increase the concentration on the regions containing high-quality solutions, and it is worth mentioning the update of the best route.

Typically, ρ refers to the random number, uniformly selected in the interval [0, 1], where the update is stored inside the pheromones matrix without being adopted to give decisions based on Equation (21).

$$\tau_j^{new} \leftarrow \rho + \tau_j^{old} + \sum \Delta\tau_j^{(k)} \quad (21)$$

Step 6 in Figure 5 represents the ascending-sort technique to obtain the best solution for each iteration (generation). According to that, the evaluation stage regarding selecting the best solution is compared with the previous one, based on Step 7 in Figure 5.

3.3.6 Selecting the Initial Best Solution

In this section, the proposed algorithm depends on the initial best solution, as in Step 8 in Figure 5, which consists of a set of routes. Each route includes a group of containers distributed between the vehicles according to the vehicle capacity, in which each container has a certain non-negative demand.

3.3.7 Eliminate Sub-Tour

The local search method eliminates the sub-tour, avoiding extra travel costs. Typically, the sub-tour includes one or two containers [42]. Step 9 in Figure 5 uses the neighbourhood operator technique, as shown in Algorithm 3.

Steps 1 - 6 in Algorithm 3 are coded to recognize the sub-tour through reading the best initial solution. As of Step 3, it is used to reverse all elements of the sub-list to the left side to check if any sub-list has one or two containers (sub-tour).

Subsequently, remove the specified sub-tour and insert it in a closed route with smaller containers based on Steps 7 - 17. After that, the algorithm will keep the new results (a) in a buffer (p), delete them from the base solution, and then merge its elements inside a sub-list with smaller containers.

Exploration techniques improve the search space [43]. Consequently, the improvement involves reducing the distance of waste vehicles during travel from source to destination. This research improves the search space by employing the Sequential Variable Neighborhood Search Change Step (SVNSCS) algorithm to explore solution space and decide which neighbourhood will be explored as the next. Moreover, some solutions will be accepted as new incumbent solutions. In other words, the search is resumed in the first neighbourhood structure of the new incumbent solution if there is a feasible solution. Otherwise, the search will continue in the next neighbourhood (according to the defined order).

Algorithm 3 Pseudocode of technique to eliminate sub-tour problem

```
1   Input. Initial best solution
2    Output. Best solution with no sub-tour.
3    Procedure subtour (BestSolution)
4       BestSolution.reverse()
5       for a := BestSolution  do # seek any sub tour in route
6           if length(a)==1 or length(a)==2 then
7               p←a
8               del BestSolution [0]
9               for b := BestSolution do
10                  if length(b)==2 then
11                      b. extend(p) # Expand sub-list
12                      BestSolution.reverse()
13                       break
14                  End if
15                  Else if length(b)==3 then
16                          b. extend(p)
17                          BestSolution.reverse()
18                          break
19              End for
20          return BestSolution
21          End if
22      End for
23  End procedure
```

The approach of the Sequential Variable Neighbourhood Search Change Step is given in Figure 4. The algorithm proposes an exchange for each pair of container locations. It moves sequentially from the first index of the sub-list to the last index in the same sub-list in terms of search space exploration until a feasible solution is found, which means it gives the shortest route to the best container locations.

Correspondingly, the proposed algorithm will return the order of the sub-list to the old route using the rotation technique to ensure the route is not corrupted. That condition is used for evaluation values of individual route distance [44]. With old individual route distance, the features of swapping each pair of containers sequentially make it useful in a narrow area of the search. Other than that, the parallel leads to accelerating the search process. Finally, the best solution will be selected based on the ascending order selection technique. The proposed algorithm stops after reaching the maximum iteration number, as in steps (10 -11) in Figure 5.

4.0 RESULTS AND DISCUSSION

The ACO-SVNSCS algorithm has been examined in terms of performance and effectiveness with several CVRP benchmark datasets with different container sizes [45, 46]. This paper has evaluated the performance of the ACO-SVNSCS algorithm based on four benchmark CVRP datasets, categorized into classes A, B, P, and E. The first three classes were authored by Augerat, whereas class E was established by Christofides and Eilon [47].

Here, we have taken the dimension as the number of containers in each dataset. The parameters of the proposed algorithm are considered based on trial and error and listed as follows: the maximum number of ants, maximum iterations number, α, β, and ρ are 50, 250, 1, 4, and 0.9, respectively. The simulation was coded in the Spyder environment (anaconda3 / 3.8 pythons), and all tests were performed on core i5 at 2.3 GHz with 4 GB RAM under the Microsoft Windows 8.1 operation system. All the simulation datasets utilized to evaluate the proposed algorithm can be found at http://vrp.atd-lab.inf.puc-rio.br/index.php/en/. In this paper, the simulation results of the proposed algorithm tested 37 instances of benchmark datasets that have solved the CVRP, as shown in Table 1.

4.1 Evaluation Methods

In the literature, several methods are used to evaluate the algorithms. The quality of the solution is measured as a difference between the optimal value of the Best-Known Solution (BKS) and the best solution value (BestSol) obtained [48, 49]. In this paper, we considered this method for evaluating each instance in the dataset category based on Equation 22.

$$\text{Percentage difference } (\Delta\%) = \frac{(\text{BestSol} - \text{BKS})}{\text{BKS}} \times 100 \quad (22)$$

The algorithm's performance can also be measured in terms of benchmark category [9]. The percentage of the average best solution can be measured based on Equation 23.

$$\text{Avg. } (\%) = \frac{\sum \text{BestSol}}{T_{ins}} \times 100 \quad (23)$$

Where

Avg.: Average of the best solution on benchmark class

T_{ins} : Total number of instances.

Table 1 Dimensions of dataset instances class

Class	Number of instances	Problem dimension
A	9	32-69
B	11	31-78
E	8	22-101
P	9	55-101
Sum	**37**	

4.2 Analysis of the Performance of ACO-SVSNCS

We performed experiments to investigate the contribution of local search improvement algorithms and ACO mechanisms to solution quality. In this research, both evaluation methods mentioned above have been considered for the experiment with the proposed algorithm's influence in minimizing travel distance. Tables 2 – 5 demonstrate the results of the comparison of ACO-SVNSCS in different benchmarks of categories A, B, E, and P. Based on that, all these tables first consist of evaluation parameters, which, in turn, involve four columns, and the first column include different dataset instances (e.g., A-n32-k5, A-n33-k5, etc.). Moreover, it has a specific number of containers distributed in scattered form, associated with demands (waste amount) for each container.

The second column represents the dimensions of the instance (problem size), and the third column is the number of vehicles (*k*). Apart from that, BKS for each instance can be obtained from the dataset. Meanwhile, the next columns indicate that the comparison between the basic ACO with the proposed algorithm in terms of best values has been minimized, in which the average best value (Avg.) and the percentage difference (Δ%) discovered an improvement value that shows the difference between the basic best value and the proposed best values. Moreover, another column was assigned to calculate the runtime in each instance to measure the proposed algorithm efficiency by calculating the tightness that represents the tolerance of the vehicle's capacity when filled by the waste load.

In Table 2, the performance of ACO-SVNSCS has been evaluated in terms of BKS. The best solution values of ACO-SVNSCS were near to the BKS value, and the best solution of ACO-SVNSCS increased (worst values) with increasing the size of the problem in special cases such as A-n46-k7, A-n54-k7, A-n63-k9, and A-n69-k9. Regarding the second evaluation method, the ACO-SVNSCS algorithm outperformed the basic ACO in 6 out of 9 instances highlighted in the bold font by reducing the number of vehicles in two instances, A-n32-k5 and A-n39-k5. This outcome will reduce the transportation cost caused by utilizing extra vehicles to service the containers. However, the asterisk symbol beside the bold font refers to the instances that reduced the number of vehicles. The ACO-SVNSCS fails in improving three instances, A-n33-k5, A-n44-k6, and A-n54-k7, which means the best solution value of the proposed algorithm obtained is similar to the best solution of the basic ACO algorithm. Eventually, the percentage value of outperforming the proposed algorithm compared to the basic algorithm is 66.7%.

Table 3 shows the evaluation of the proposed algorithm in benchmark B. ACO-SVNSCS is near to finding BKS in 9 out of 11 instances where it succeeds in improving the best solution at the rate of 81.81%. Apart from that, the number of vehicles has been reduced only in one instance, B-n38-k6. In comparison, ACO-SVNSCS fails to optimize the solution in two instances, represented by B-n31-k5 and B-n50-k7.

Table 4 presents the evaluation of the proposed algorithm in benchmark E. The ACO-SVNSCS algorithm also succeeds in improving the best solution. With the same number of vehicles (*K*) at the average of 62.5%, finding the best solution near BKS is comprised of 5 out of 8 instances. Still, three instances, E-n22-k4, E-n33- k4 and E-n76-k8, did not optimize.

Table 2 Results of comparison between basic ACO algorithm with a proposed algorithm based on benchmark A

Evaluation parameters				ACO			ACO-SVNSCS					
Instances	**Dim**	**K**	**BKS**	**Best**	**Avg.**	**(Δ%)**	**Best**	**Avg.**	**(Δ%)**	**Imp.**	**Runtime (s)**	**Tig.**
A-n32-k5	32	5	784	846	846	7.9	**785***	794.7	0.1	61	100.3	1.0
A-n33-k5	33	5	661	681	681	3.0	681	696.8	3.0	0	45.2	0.89
A-n33-k6	33	6	742	761	762.73	2.6	**761.0**	813.8	2.6	0.22	66.7	0.75
A-n39-k5	39	6	831	882	884	6.1	**867***	867	4.3	15	98.55	1.0
A-n44-k6	44	6	937	975	984.19	4.1	975	975	4.1	0	172.2	0.95
A-n46-k7	46	7	914	1014	1025.0	10.9	**1007**	1045.1	10.2	7	96.02	0.86
A-n54-k7	54	7	1167	1256	1256	7.6	1256	1256	7.6	0	183.5	0.95
A-n63-k9	63	9	1616	1756	1777.5	8.7	**1742**	1770.5	7.8	14	127.3	0.97
A-n69-k9	69	9	1159	1264	1279.2	9.1	**1242**	1242	7.2	22	246.7	0.93
count							6					
Avg.							66.7 %					

Tig. Tightness; **Imp.** Improvement value

Table 3 Results of comparison between basic ACO algorithm with a proposed algorithm based on benchmark B

Evaluation parameters				ACO			ACO- SVNSCS					
Instances	**Dim**	**K**	**BKS**	**Best**	**Avg.**	**(Δ%)**	**Best**	**Avg.**	**(Δ%)**	**Imp.**	**Runtime (s)**	**Tig.**
B-n31-k5	31	5	672	679	680.4	1.06	679	681	1.16	0	88.01	0.82
B-n38-k6	38	6	805	843	836.8	4.73	**783***	874	-2.66	59.4	65.8	0.85
B-n45-k5	45	5	751	788	790.13	4.93	**762**	762	1.46	26	150.01	0.97
B-n43-k6	43	6	742	755	761	1.75	**754**	756	1.62	1	167.7	0.86
B-n44-k7	44	7	909	967	979	6.38	**966**	966	6.27	1	79.07	0.91
B-n51-k7	51	7	1032	1085	1089	5.14	**1084**	1087.8	5.04	1	129.32	0.97
B-n50-k7	50	7	741	798	805	7.69	798	858.4	7.69	0	19.3	0.86
B-n52-k7	52	7	747	779	787	4.28	**776**	776.7	3.88	3	20.1	0.86
B-n56-k7	56	7	707	768	775	8.63	**767**	777	8.49	1	22.3	0.88
B-n66-k9	66	9	1316	1422	1431	8.05	**1398**	1398	6.23	24	27.7	0.95
B-n78-k10	78	10	1221	1331	1343	9.05	**1328**	1334.2	8.76	3.5	35.6	0.93
count							9					
Avg.							81.81%					

Tig. Tightness; **Imp.** Improvement value

Table 4 Results of comparison between basic ACO algorithm with a proposed algorithm based on benchmark E

Evaluation parameters				ACO			ACO- SVNSCS					
Instances	**Dim**	**K**	**BKS**	**Best**	**Avg.**	**(Δ%)**	**Best**	**Avg.**	**(Δ%)**	**Imp.**	**Runtime (s)**	**Tig.**
E-n22-k4	22	4	375	401	401.59	6.9	401	431.42	6.9	0	44.72	0.93
E-n33-k4	33	4	835	874	881.04	4.7	874	874	4.7	0	11.17	0.91
E-n51-k5	51	5	521	611	624.33	17.3	**604**	604.68	15.9	7	128.9	0.97
E-n76-k7	76	7	682	797	819	16.9	**796**	796.14	16.7	1	33.77	0.88
E-n76-k8	76	8	735	855	863.1	16.3	855	855.35	16.3	0	44.73	0.94
E-n76-k10	76	10	830	931	958.2	12.2	**921**	932.1	11.0	10	32.67	0.97
E-n76-k14	76	14	1021	1119	1133.4	9.6	**1109**	1139.1	8.6	10	214.32	0.9
E-n101-k8	101	8	817	1000	1017.69	22.4	**999.69**	999.82	22.4	0.31	49.88	0.91
count							5					
Avg.							62.5%					

Tig. Tightness; **Imp.** Improvement value

Table 5 Results of comparison between basic ACO algorithm with a proposed algorithm based on benchmark P

Evaluation parameters				ACO			ACO- SVNSCS					
Instances	**Dim**	**K**	**BKS**	**Best**	**Avg.**	**(Δ%)**	**Best**	**Avg.**	**(Δ%)**	**Imp.**	**Runtime (s)**	**Tig.**
p-n55-k15	55	15	989	999	1016.5	1.01	**988****	988.4	-0.1	11	21.75	0.93
p-n60-k10	60	10	744	830.23	838.41	11.59	**787**	790.46	5.8	43.2	23.90	0.94
p-n60-k15	60	15	968	1042	1056.1	7.64	**1016**	1026.2	5.0	26	24.42	0.94
P-n45-k5	45	5	510	577	587.3	13.14	577	600.5	13.1	0	21.9	0.92
P-n65-k10	65	10	792	889	908.9	12.25	**871**	874.8	10.0	18	26.61	0.93
P-n76-k4	76	4	593	690	701.7	16.36	**678**	678.35	14.3	12	38.55	0.94
P-n70-k10	70	10	827	931.6	949.44	12.65	**930.1**	944.9	12.5	1.5	35.42	0.97
P-n76-k5	76	5	627	733	754.9	16.91	**727**	727.7	15.9	6	41.85	0.97
p-n101-k4	101	4	681	821	840.1	20.56	821	834.2	20.6	0	240	0.91
count							7					
Avg.							77.7%					

Tig. Tightness; **Imp.** Improvement value

Figure 6 Improvement for each instance in terms of benchmark: (a) Class A, (b) Class B, (c) Class E, (d) Class P

For benchmark P, Table 5 shows that the improvement is 77.77%, involving 7 out of 9. However, in one of the instances, p-n55-k15 increased by one vehicle, and the ACO-SVNSCS fails in an improved solution for two instances, P-n45-k5 and P-n101-k4, compared to the basic ACO. Otherwise, it was characterized by a good improvement in the instance P-n60-k10.

4.3 Performance Improvement

The improvement values (Imp) can be calculated by the difference between the best value of the basic ACO algorithm ($f(S)$) and the best solution of the proposed algorithm ($f(S')$), as shown in Equation (24).

$$\text{Imp} = f(S) - f(S') \qquad (24)$$

The improvement values of the proposed algorithm have been superior statistically over the basic ACO, as shown in Figure 6(a), (b), (c), and (d).

Concerning performance analysis for the algorithm proposed, it clearly shows that the improvements outperformed the basic ACO algorithm, as shown in Table 6.

Table 6 Percentage of Improvement for ACO- SVNSCS

Benchmarks	Improvement (%)
A	66.7
B	81.81
E	62.5
P	77.77

5.0 CONCLUSION AND FUTURE DIRECTION

Estimating and controlling waste collection costs is important in setting up a cost-effective system. It has become a critical problem in most local waste management companies that aspire to provide the best services to their citizens. This paper has adopted a new Sequential Variable Neighborhood Search Change Step with Ant Colony Optimization (ACO-SVNSCS) to minimize route distance and the number of vehicles used for service containers in a certain network to solve the capacitated vehicle routing problem. A comparative analysis was made between the proposed algorithm and standard ACO. The proposed algorithm succeeds in solving small, medium, and large-scale problems. From observation, in the instances containing a single sub-tour, such as A-n32-k5 and A-n39-k5, the distance improvement will be large because the number of vehicles that served containers has been reduced by eliminating the sub-tour, leading to a reduction in transportation costs and workers' wages. Another finding was the difficulty in the working strategy of basic ACO, which happens when the algorithm starts choosing the first container at the beginning of each route. ACO-SVNSCS is a competitive algorithm in terms of improving the basic ACO by enhancing the exploration of the search space by addressing problems in 6, 9, 5, and 7 instances out of 9, 11, 8, and 10 with improvement around 66.7%, 81.81%, 62.5% and 77.77% for dataset A, B, E, and P, respectively.

The tightness value was very close in all instances, which may be seen at http://vrp.atd-lab.inf.puc-rio.br/index.php/en/. Concerning future directions, a hybridization of the ACO algorithm with the same Sequential Variable Neighborhood Search Change Step (SVNSCS) algorithm or with other Variable Neighborhood Searches (VNS), such as Cyclic neighbourhood change step, Pipe neighbourhood change step, and the skewed neighbourhood change step can be employed to measure the run time and the convergence of the proposed algorithm for proving the efficiency.

Acknowledgment

The Ministry of Higher Education Malaysia partially supported this work under the Fundamental Research Grant Scheme: FRGS/1/2019/TK04/USM/02/12.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] Akhtar, M., Hannan, M. A., Basri, H., and Scavino, E. 2015. Solid Waste Generation and Collection Efficiencies: Issues and Challenges. *Jurnal Teknologi.* 75(11). Doi: https://doi.org/10.11113/jt.v75.5331.

[2] Ismail, N., and Ani, F. N. 2015. A Review on Plasma Treatment for the Processing of Solid Waste. *Jurnal Teknologi.* 72(5). Doi: https://doi.org/10.11113/jt.v72.3938.

[3] Munyai, O. and Nunu, W.N. 2020. Health Effects Associated with Proximity to Waste Collection Points in Beitbridge Municipality, Zimbabwe. *Waste Management*. 105: 501–510. Doi: https://doi.org/10.1016/j.wasman.2020.02.041.

[4] Jiang, P., Van Fan, Y., Zhou, J., Zheng, M., Liu, X. and Klemeš, J. J. 2020. Data-driven Analytical Framework for Waste-dumping Behaviour Analysis to Facilitate Policy Regulations. *Waste Management*. 103: 285–295. Doi: https://doi.org/10.1016/j.wasman.2019.12.041.

[5] Hoque, M. M. and Rahman, M. T. U. 2020. Landfill Area Estimation based on Solid Waste Collection Prediction using ANN Model and Final Waste Disposal Options. *Journal of Cleaner Production.* 256: 120387. Doi: https://doi.org/10.1016/j.jclepro.2020.120387.

[6] Samsudin, M. D. M. and Don, M. M. 2013. Municipal Solid Waste Management in Malaysia: Current Practices, Challenges, and Prospects. *Jurnal Teknologi.* 62(1). Doi: https://doi.org/10.11113/jt.v62.1293.

[7] Liman, A. and Ngah, I. 2015. Appraisal of Municipal Solid Waste Management (MSWM) in Jimeta-Yola Adamawa State of Nigeria: The Need for Innovative, PPP Strategies. *Jurnal Teknologi.* 77(15). Doi: https://doi.org/10.11113/jt.v77.6537.

[8] Bees, A. D. and Williams, I. D. 2017. Explaining the Differences in Household Food Waste Collection and Treatment Provisions between Local Authorities in England and Wales. *Waste Management.* 70: 222–235. Doi: https://doi.org/10.1016/j.wasman.2017.09.004.

[9] Hannan, M. A., Akhtar, M., Begum, R. A., Basri, H., Hussain, A. and Scavino, E. 2018. Capacitated Vehicle-routing Problem Model for Scheduled Solid Waste Collection and Route Optimization using PSO Agorithm. *Waste management*. 71: 31–41. Doi: https://doi.org/10.1016/j.wasman.2017.10.019.

[10] Foday, Jr, E. H., Ramli, N. A. S., Ismail, H. N., Malik, N. A., Basri, H. F., Aziz, F. S. A., Nor, N. S.M. and Jumhat, F. 2017. Municipal Solid Waste Characteri stics in Taman Universiti, Skudai, Johore, Malaysia. *Journal of Advanced Research Design.* 38(1): 13–20.

[11] Nasir, S. M., Othman, N. H., Isa, C. M. and Ibrahim, C. C. 2016. The Challenges of Construction Waste Management in Kuala Lumpur. *Jurnal Teknologi*. 78: 5–3. Doi: https://doi.org/10.11113/jt.v78.8522.

[12] Sahib, F. S. and Hadi, N. S. 2021. Truck Route Optimization in Karbala City for Solid Waste Collection. *Materials Today: Proceedings*. Doi: https://doi.org/10.1016/j.matpr.2021.06.394.

[13] Tirkolaee, E. B., Mahdavi, I. and Esfahani, M. M. S. 2018. A Robust Periodic Capacitated Arc Routing Problem for Urban Waste Collection Considering Drivers and Crew's Working Time. *Waste Management*. 76: 138–146. Doi: https://doi.org/10.1016/j.wasman.2018.03.015.

[14] Sahib, T. M., Mohd-Mokhtar, R., Alharan, A. F. H., Ali, N. S., & Kadhum, A. A. 2024. An Improved Strategy for Solid Waste Management based on Programmable Logic Controller. *Int. J. of Environment & Waste Management*. 34(3): 297–310. https://doi.org/10.1504/IJEWM.2024.139963.

[15] Louati, A. 2016. Modeling Municipal Solid Waste Collection: A Generalized Vehicle Routing Model with Multiple Transfer Stations, Gather Sites and Inhomogeneous Vehicles in Time Windows. *Waste Management*. 52: 34–49. Doi: https://doi.org/10.1016/j.wasman.2016.03.041.

[16] Armington, W. R. and Chen, R. B. 2018. Household Food Waste Collection: Building Service Networks through Neighbourhood Expansion. *Waste Management.* 77: 304–311. Doi: https://doi.org/10.1016/j.wasman.2018.04.012.

[17] Han, H. and Ponce Cueto, E. 2015. Waste Collection Vehicle Routing Problem: Literature Review. *PROMET-Traffic & Transportation*. 27(4): 345–358. Doi: https://doi.org/10.7307/ptt.v27i4.1616.

[18] Vu, H. L., Ng, K. T. W., Fallah, B., Richter, A. and Kabir, G. 2020. Interactions of Residential Waste Composition and Collection Truck Compartment Design on GIS Route Optimization. *Waste Management*. 102: 613–623. Doi: https://doi.org/10.1016/j.wasman.2019.11.028.

[19] Akpinar. S. 2016. Hybrid Large Neighbourhood Search Algorithm for Capacitated Vehicle Routing Problem. *Expert Systems with Applications.* 61: 28–38. Doi: https://doi.org/10.1016/j.eswa.2016.05.023.

[20] Imran. A, Luis. M. and Okdinawati. L. 2016. A Variable Neighborhood Search for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Jurnal Teknologi.* 78(9): 53–58. Doi: https://doi.org/10.11113/jt.v78.6062.

[21] İlhan, İ. L. H. A. N. 2021. An Improved Simulated Annealing Algorithm with Crossover Operator for Capacitated Vehicle Routing Problem. *Swarm and Evolutionary Computation.* 64: 100911. DOI: https://doi.org/10.1016/j.swevo.2021.100911.

[22] Dalbah, L. M., Al-Betar, M. A., Awadallah, M. A. and Zitar, R. A. 2022. A Modified Coronavirus Herd Immunity Optimizer for Capacitated Vehicle Routing Problem. *Journal of King Saud University-Computer and Information Sciences.* 34(8): 4782–4795. Doi: https://doi.org/10.1016/j.jksuci.2021.06.013.

[23] Sahib, T. M., Mohd-Mokhtar, & R., Kassim, A. M. 2022. Survey on Meta-Heuristic Algorithms for Solving Vehicle Route Problems in a Waste Collection System. In: Mahyuddin N. M., Mat Noor N. R., Mat Sakim H. A. (Eds). Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 829. Springer, Singapore. https://doi.org/https://doi.org/10.1007/978-981-16-8129-5_57.

[24] Tirkolaee, E. B., Alinaghian, M., Bakhshi Sasi, M. and Seyyed Esfahani, M. M. 2016. Solving a Robust Capacitated Arc Routing Problem Using a Hybrid Simulated Annealing Algorithm: A Waste Collection Application. *Journal of Industrial Engineering and Management Studies*. 3(1): 61–76.

[25] Tirkolaee, E. B., Alinaghian, M., Hosseinabadi, A. A. R., Sasi, M. B. and Sangaiah, A. K. 2019. An Improved Ant Colony Optimization for the Multi-trip Capacitated Arc Routing Problem. *Computers & Electrical Engineering*. 77: 457–470. Doi: https://doi.org/10.1016/j.compeleceng.2018.01.040.

[26] Raflesia, S. P. and Pamosoaji, A. K. 2019, November. A Novel Ant Colony Optimization Algorithm for Waste Collection Problem. *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE) on IEEE*. 413–416. Doi: https://doi.org/10.1109/ICITISEE48480.2019.9003896.

[27] Delgado-Antequera, L., Caballero, R., Sánchez-Oro, J., Colmenar, J. M. and Martí, R. 2020. Iterated Greedy with Variable Neighborhood Search for a Multiobjective Waste Collection Problem. *Expert Systems with Applications*. 145: 113101. Doi: https://doi.org/10.1016/j.eswa.2019.113101.

[28] Máximo, V. R. and Nascimento, M. C. 2021. A Hybrid Adaptive Iterated Local Search with Diversification Control to the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*. 294(3): 1108–1119. Doi: https://doi.org/10.1016/j.ejor.2021.02.024.

[29] Akhtar, M., Hannan, M. A., Begum, R. A., Basri, H. and Scavino, E. 2017. Backtracking Search Algorithm in CVRP Models for Efficient Solid Waste Collection and Route Optimization. *Waste Management*. 61: 117–128. Doi: https://doi.org/10.1016/j.wasman.2017.01.022.

[30] Jorge, D., Antunes, A. P., Ramos, T. R. P. and Barbosa-Póvoa, A. P. 2022. A Hybrid Metaheuristic for Smart Waste Collection Problems with Workload Concerns. *Computers & Operations Research*. 137: 105518. Doi: https://doi.org/10.1016/j.cor.2021.105518.

[31] Wy, J. and Kim, B. I. 2013. A Hybrid Metaheuristic Approach for the Rollon–rolloff Vehicle Routing Problem. *Computers & Operations Research*. 40(8): 1947–1952. Doi: https://doi.org/10.1016/j.cor.2013.03.006.

[32] Jabir, E., Panicker, V. V., & Sridharan, R. 2017. Design and Development of a Hybrid Ant Colony-variable Neighbourhood Search Algorithm for a Multi-depot Green Vehicle Routing Problem. *Transportation Research Part D: Transport and Environment*. 57: 422–457. https://doi.org/10.1016/j.trd.2017.09.003.

[33] Hansen, P., Mladenović, N., Todosijević, R. and Hanafi, S. 2017. Variable Neighborhood Search: Basics and Variants. *EURO Journal on Computational Optimization*. 5(3): 423–454. Doi: https://doi.org/10.1007/s13675-016-0075-x.

[34] Yassen, E. T., Ayob, M., Nazri, M. Z. A. and Sabar, N. R. 2017. An Adaptive Hybrid Algorithm for Vehicle Routing Problems with Time Windows. *Computers & Industrial Engineering*. 113: 382–391. Doi: https://doi.org/10.1016/j.cie.2017.09.034.

[35] Souza, I. P., Boeres, M. C. S., & Moraes, R. E. N. 2023. A Robust Algorithm based on Differential Evolution with Local Search for the Capacitated Vehicle Routing Problem. *Swarm and Evolutionary Computation*. 77: 101245. https://doi.org/10.1016/j.swevo.2023.101245.

[36] Ali. H and Kar. A. K. 2018. Discriminant Analysis using Ant Colony Optimization - An Intra-Algorithm Exploration. Procedia Computer Science. 132: 880–889. Doi: https://doi.org/10.1016/j.procs.2018.05.100.

[37] Pedemonte. M., Nesmachnow. S. and Cancela. H. 2011. A Survey on Parallel Ant Colony Optimization. *Applied Soft Computing Journal*. 11(8): 5181–5197. Doi: https://doi.org/10.1016/j.asoc.2011.05.042.

[38] Ebadinezhad. S. 2020. DEACO: Adopting Dynamic Evaporation Strategy to Enhance ACO Algorithm for the Traveling Salesman Problem. *Engineering Applications of Artificial Intelligence*. 92: 103649. Doi: https://doi.org/10.1016/j.engappai.2020.103649.

[39] Faiz, S., Krichen, S. and Inoubli, W. 2014. A DSS based on GIS and Tabu Search for Solving the CVRP: The Tunisian Case. *The Egyptian Journal of Remote Sensing and Space Science*. 17(1): 105–110. Doi: https://doi.org/10.1016/j.ejrs.2013.10.001.

[40 Huang, S. H., Huang, Y. H., Blazquez, C. A. and Paredes-Belmar, G. 2018. Application of the Ant Colony Optimization in the Resolution of the Bridge Inspection Routing Problem. *Applied Soft Computing*. 65: 443–461. Doi: https://doi.org/10.1016/j.asoc.2018.01.034.

[41] Guo, N., Qian, B., Hu, R., Jin, H. P. and Xiang, F. H. 2020. A Hybrid Ant Colony Optimization Algorithm for Multi-Compartment Vehicle Routing Problem. *Complexity*. 2020. Doi: https://doi.org/10.1155/2020/8839526.

[42] Tang, J., Zhang, J. and Pan, Z. 2010. A Scatter Search Algorithm for Solving Vehicle Routing Problem with Loading Cost. *Expert Systems with Applications*. 37(6): 4073–4083. Doi: https://doi.org/10.1016/j.eswa.2009.11.027.

[43] Buhrkal, K., Larsen, A. and Ropke, S. 2012. The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context. *Procedia-Social and Behavioral Sciences*. 39: 241–254. Doi: https://doi.org/10.1016/j.sbspro.2012.03.105.

[44] Zhang, D., Dong, R., Si, Y.W., Ye, F. and Cai, Q. 2018. A Hybrid Swarm Algorithm based on ABC and AIS for 2L-HFCVRP. *Applied Soft Computing*. 64: 468–479. Doi: https://doi.org/10.1016/j.asoc.2017.12.012.

[45] Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T. and Subramanian, A. 2017. New Benchmark Instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*. 257(3): 845–858. Doi: https://doi.org/10.1016/j.ejor.2016.08.012.

[46] Kyriakakis, N.A., Marinaki, M. and Marinakis, Y. 2021. A Hybrid Ant Colony Optimization-Variable Neighborhood Descent Approach for the Cumulative Capacitated Vehicle Routing Problem. *Computers & Operations Research*. 134: 105397. Doi: https://doi.org/10.1016/j.cor.2021.105397.

[47] Matthopoulos, P. P., & Sofianopoulou, S. 2019. A Firefly Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *International Journal of Industrial and Systems Engineering*. 33(2): 204–224.

[48] Arnold, F., Santana, Í., Sörensen, K. and Vidal, T. 2021. PILS: Exploring High-order Neighborhoods by Pattern Mining and Injection. *Pattern Recognition*. 116: 107957. Doi: https://doi.org/10.1016/j.patcog.2021.107957.

[49] Gunduz. M. and Aslan. M. 2021. DJAYA: A Discrete Jaya Algorithm for Solving Traveling Salesman Problem. *Applied Soft Computing*. 105: 107275. Doi: https://doi.org/10.1016/j.asoc.2021.107275.