Teknomekanik, Vol. 7, No. 1, pp. 38-61, June 2024
e-ISSN: 2621-8720 p-ISSN: 2621-9980

A comparative study utilizing hybridized ant colony optimization algorithms for solving dynamic capacity of vehicle routing problems in waste collection system

Thaeer Mueen Sahib[1,2], Rosmiwati Mohd-Mokhtar[1*] and Azleena Mohd-Kassim[3]

[1] School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, Pulau Pinang, **MALAYSIA**
[2] Department of Electrical Techniques, Kufa Technical Institute, Al-Furat Al-Awsat Technical University, **IRAQ**
[3] School of Computer Science, Universiti Sains Malaysia, Pulau Pinang, **MALAYSIA**

* Corresponding Author: eerosmiwati@usm.my
Received December 29th 2023; 1st Revised May 01st 2024; 2nd Revised May 20th 2024; Accepted May 24th 2024

Cite this https://doi.org/10.24036/teknomekanik.v7i1.28872

Abstract: The waste collection stage generated 70% of the cost of the total Municipal Solid Waste (MSW) management system. Therefore, choosing the most affordable waste collection method is essential to accurately estimate the waste collection and transportation costs, thus selecting the required vehicle capacity. The study aims to design a waste collection system for calculating waste collection and transportation costs using a systematic framework that includes Hybridized Ant Colony Optimization (HACO) with Sequential Variable Neighborhood Search Change Step (SVNSCS) and Sequential Variable Neighborhood Decent (SVND). The framework addresses a Dynamic Capacity of Vehicle Routing Problem (DCVRP) and improves ACO's ability in exploration and exploitation stages. The objectives are to minimize the cost of travel distance and arrival time formulated in a mathematical model and to design a new strategy for eliminating the sub-tour problem in the following steps: (1) minimize the number of routes assigned, (2) increase the amount of waste in the vehicle capacity, and (3) define the best amount of waste allowed in vehicle capacity. The waste collection system compared HACO with ACO across four benchmark datasets. The results indicate HACO outperformance ACO at 100%, 91%, 100%, and 87%, respectively. The visualization results demonstrated that HACO has fast convergence and can be considered another essential tool for route optimization in the waste collection system.

Keywords: Ant colony optimization; Dynamic capacity of vehicle routing problem; Sequential variable neighborhood search change step; Waste collection

1. Introduction

Recent studies show that most local governments in developing nations still struggle with managing MSW [1], [2]. MSW involves several activities, including production, storage, disposal, treatment, recycling, collecting, and transportation to a transfer station before being dumped (landfill site) [3], [4]. Compared to other operations, the Solid Waste Collection (SWC) expense and transportation stage consume over 70% of the budget [5]. A study by Munyai and Nunu stated the increase in waste generation from 1.5 billion tons to 3.4 billion tons by 2025 worldwide [6]. In addition, a study by Silva et al. [7] confirmed about 2.01 billion tons of MSW annually, which is expected to reach 3.4 billion tons in 2050 globally. The increase in waste generation triggers serious action for appropriate needs and demands for good waste collection systems.

Poorly managed waste causes the clogging of sewage channels, consequently flooding roads and spreading diseases [8], [9]. Moreover, an improper segregation system and inadequate waste vehicles make the problem more complicated and challenging due to increased availability and quantity, which also deteriorates the environmental aspect [10]. The use of high-capacity vehicles to transport waste from the inner city's periphery to recycling centers outside the city is an additional concern [11]. Furthermore, fuel efficiency is significantly impacted by vehicle type, load weight, road inclination, and the distance traveled, which is a critical factor in determining fuel consumption [12]. However, due to capacity constraints and manual operation, these vehicles can only collect compact and dense waste materials, such as the contents of streetside garbage bins [13].

Developing a software tool is necessary to manage efficient waste collection and sustainability [9]. Maximizing the profit of waste quantity can be done by eliminating sub-tour costs, which reduces the number of vehicles used. A typical example would be either a controller system method to reduce the time required for waste disposal or a vehicle routing method as a management system [14], [15]. These studies examined how vehicle capacity dynamics affected the vehicle routing problem and the best route selection in the waste collection system. Some variations, for example, minimizing route distance, strongly correlate with lowering waste collection costs [16].

This research encompasses multiple objective functions. The motivation to maximize the profit of waste quantity can be achieved by eliminating sub-tour costs, which reduces the number of vehicles used; consequently, the economic aspect increased. Furthermore, the task involves exploring optimal strategies (focusing on distance reduction) for efficient waste collection. This entails considering the geographical coordinates of collection points (specifically commercial sites) and the corresponding waste quantities generated within a defined geographical region. The next section discusses the variants in the literature concerning vehicle routing problems. It will define the research gap and highlight the proposed solution strategies for addressing the problem.

2. Literature review

Numerous studies have concentrated on using heuristic and metaheuristic algorithms to enhance waste processes [15]. Earlier studies aimed to minimize the cost of travel distance, time, and emissions with variables in the Vehicle Routing Problem (VRP) [17]. In the waste collection approach, the VRP has several variants in which the waste is picked up from a set of CPs and delivered to the destination. The variants can be subjected to a time window with a homogenous or heterogeneous vehicle fleet [18]. The Capacity of Vehicle Routing Problem (CVRP) also deals with a set of nodes or arcs and is known as the Capacity of Arc in Routing Problem (CARP) [19], [20]. As a result, the solid waste should be collected from the CPs, keeping in mind that the vehicle's capacity will decrease as it moves from one node to another.

Henke et al. [21] proposed a study focused on utilizing a multi-compartment exclusively for classification waste purposes and determining the number of compartments the vehicle capacity can be divided and the size of each compartment. Along with other optimization problems, Li et al. [22] employed a Fuzzy Stochastic Quadratic Programming (SFQP) method for the best waste management to minimize collection and transportation costs. Note that the estimation costs depend on container size, cyclic waste collection, and collection time in each load. In addition, this technique has several complexities of mathematics. Ghiani et al. [23] utilized the Integer Linear Programming (ILP) model in two phases. The first phase combines heuristic with metaheuristic algorithms for the waste collection stage, and the second phase represents the heuristic approach for the waste transportation stage. The research employed a small vehicle capacity and aimed to decrease the number of vehicles used by incorporating the collection zone into the existing route, removing it from the collection zone, and revising the residual capacity and time.

Another study employed two local searches involving hill climbing and tabu search to improve the features of the initial solution in terms of minimizing the total routing time, which was assigned for the Sectoring-Arc Routing Problem (SARP) in residential waste collection [22]. Additionally, Ismail et al. [24] proposed a basic ACO algorithm to solve the travel salesman problem to minimize the route distance in a single route. This study has no evaluation to ensure the performance of the proposed methods. Nevertheless, the mathematical model applies to minor problems.

Edwards et al. [25] verified that the constraints on MSW collection depend on the natural geography, traffic congestion, and inadequate transport infrastructures. For example, two studies conducted in Iraq demonstrated that traditional methods were still used for collecting and transporting waste [26], [27]. In the same context, a study conducted by Mat et al. [28] focused on reducing the number of vehicles used by solving a dynamic speed for the VRP. From the study, the dynamic travel speed of vehicles in a waste collection problem greatly affects the total travel time, total travel distance and the number of vehicles needed. According to the related work in the literature, Table 1 summarizes the analysis of most variants' problems and specifies which variants are still not presented, which inspired the current research.

Table 1. Literature on Solid Waste Collection

Ref.	Optimization														
	Single Objective	Multi-Objective	Min	Max	Capacitated of nodes (CVRP)	Capacitated of edges (CARP)	Multi-compartment vehicle routing problem (MCVRP)	Dynamic of CVRP (DCVRP)	Vehicle fixed speed	Vehicle dynamic speed	Distance covered	Computational time	Threshold waste level in containers	Tightness	Time constraint (VRPTW)
[14]	√	×	√	×	√	×	×	×	×	×	√	×	×	×	√
[16]	√	×	√	×	√	×	×	×	×	×	√	√	√	√	×
[17]	√	×	√	×	√	×	√	×	×	×	√	√	×	×	×
[18]	√	×	√	×	×	×	×	×	×	×	√	×	×	×	√
[20]	√	×	√	×	×	√	×	×	√	×	√	×	×	×	√
[21]	√	×	√	×	×	√	×	×	×	×	√	√	×	×	√
[22]	√	×	√	×	×	×	×	×	×	×	√	×	×	×	×
[24]	√	×	√	×	×	×	×	×	×	×	√	×	×	×	√
[26]	√	×	√	×	×	×	×	×	×	√	√	×	×	×	√
This study	√	√	√	√	√	×	×	√	√	×	√	√	×	√	√

Based on the literature review in Table 1, no study focused on solving the dynamic capacitated vehicle routing problem (DCVRP) to find optimal routes and lower the number of vehicles used to service all containers in a specific zone. Therefore, this study addresses this issue by hybridizing the ACO with sequential improvement procedures. This hybridized approach enhances the initial solution of the ACO algorithm, thus increasing the waste collection (WC) system's performance. Subsequently, the objectives of the study are presented as follows:

1) To design a new systematic framework model by hybridizing ACO to increase its searchability in the exploration and exploitation stages.
2) To design a new strategy to eliminate the sub-tour problem, thus reducing the number of routes to be collected.
3) To identify the average vehicle speed variant and determine the arrival time at each CP.
4) To analyze the effect of dynamic vehicle capacity on the VRP by adopting a novel vehicle capacity threshold.
5) To evaluate the performance of the HACO with the Best-Known Solution (BKS) and its sensitivity in terms of the dynamic capacity of VRP.

3. Methods

This section comprises of six essential stages: First, developing the theoretical foundations of the DCVRP problem using a homogeneous vehicles approach. The flowchart encompasses the hybridization of the proposed algorithm as well as the methods and techniques employed to enhance its performance, while also providing an overview of the optimization system's framework. Then, an explanation of the initial solution structure model and the details of the solution's parameters are discussed. Next, the operational mechanisms of the solution's demolition and construction model are clarified. Finally, an explanation of how the model determines the optimal solutions from among the iterations that are available. Moreover, in the case of heterogeneous vehicles, the theoretical framework underlying DCVRP and the work schedule for the technique that determines the optimal vehicle capacity for waste disposal is described.

3.1 Formulation of the DCVRP model

The DCVRP can be formulated schematically as $G = (N, E)$, where *G* represents the graph. Here, *N* represents a set of containers graphically distributed, $N = 0, 1 \dots, n$, where the depot index is 0, and the other index represents the containers ranging from 1 to *n*. Meanwhile, *E* represents a set of edges called $V = 0, 1, \dots, n$, where each edge $(i, j) \in E = \{(i, j): i, j \in V, i \neq j\}$ [29]. The edge weights indicate the travel time t_{ij} and edge distance d_{ij} for each edge between two containers where $(i \neq j)$. Each container $i \in V' = V \setminus \{0\}$ has a unique demand q_i refer to the amount of waste for $i = 1, 2, \dots, n$. The depot is the starting place for homogenous vehicles $k = 1, 2 \dots K$, that serve each container without exceeding the capacity threshold of the vehicle $Q^k$and return to the same depot upon completion of their route. This research will eventually analyze the dynamic influence of waste level on truck capacity while concurrently specifying the optimal route.

The formulation of the entire waste collection system must be governed by a model with three decision variables dependent on vehicle capacity, Q^k and the amount of waste in each container. The first decision variable is $X_{ijk} = 1$ if the vehicle *k* travels from container (*i*) to container (*j*) under the threshold of vehicle capacity. Otherwise, $X_{ijk} = 0$, implying that the vehicle does not travel on the scheduled route, as shown in Eq. 1.

$$X_{ijk} \begin{cases} 1, \text{if the vehicle served the edge } (i,j), \text{and } Total\ q_r \leq T_{kho} \\ 0, \ \text{Otherwise} \end{cases} \tag{1}$$

where,

T_{kho}: threshold levels of homogenous vehicle capacity.
$Total\ q_r$: total amount of waste for each route.

The second decision variable, $u_k = 1$ if the vehicle has been maintained and is ready for usage by the driver and laborers. Otherwise $u_k = 0$, and in this situation, the vehicle cannot be used based on Eq. 2.

$$u_k = \begin{cases} 1, \ \text{If } k^{th} \text{ vehicle used has been specified} \\ 0, \ \text{otherwise} \end{cases} \tag{2}$$

$$Y_{jk} = \begin{cases} 1, \text{if container i is visited by the vehicle } k^{th} \\ 0, \ \text{otherwise} \end{cases} \tag{3}$$

The objective function to minimize the total distance for waste collection, *Z,* is defined in Eq. 4.

$$Z = \min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} d_{ij} \,.X_{ijK} + \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} u_c \,.u_k + \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} t_{ij} \,.X_{ijK} \quad (4)$$

Where the first term, d_{ij} is used to calculate the total distance travelled concerning the decision variable X_{ijk}. Here, d_{ij} is a symmetrical matrix $(n \times n)$, as shown in Eq. 5 [30].

$$d_{ij} = \begin{bmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{12} & 0 & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1n} & d_{2n} & \dots & d_{nn} \end{bmatrix} \quad (5)$$

The second term u_c is assigned for determining the cost of the vehicle when the choice variable u_k is considered. Meanwhile, the third term t_{ij} is used to calculate the cost of the travel time between containers (*i*) and (*j*), which is directly proportional to the Average Drive Speed (ADS) and the distance travelled, d_{ij}. Regarding the second objective function, the profit model can be applied to maximize the amount of waste collected, q_j by the vehicle, as expressed in Eq. 6.

$$\text{Profit}, P = \max \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} q_j \,.X_{ijK} > 0, j \in \{2, \dots n\} \quad (6)$$

The model of eliminating the sub-tour problem depends directly on the total amount of q_j and the number of containers (n) in each route. Hence, this model will be described fully in the section on resolving the sub-tour problem. The following constraints are considered when making the DCVRP model.

$$\sum_{j=1}^{n} \sum_{k=1}^{K} X_{0jk} = 1 \quad (7)$$

$$\sum_{j=1}^{n} q_{0jk} = 0, \quad \forall\, k = 1, 2 \dots K \quad (8)$$

$$\sum_{i=1}^{n} \sum_{k=1}^{K} X_{i0k} = 1 \quad (9)$$

$$\sum_{k=1}^{K} X_{ijk} = \sum_{k=1}^{K} X_{jik} = Y_j \quad (10)$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} q_i \,.X_{ijk} \leq T_c \quad \forall i = 1, 2 \dots n \quad \forall j = 1, 2 \dots n \quad (11)$$

Where T_c is the threshold or the maximum vehicle capacity permitted to transport waste.

$$dist_{ij} = dist_{ji}, \forall i = 1, 2 \dots n, \forall j = 1, 2 \dots n \quad (12)$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} X_{ijk} \leq u_k \quad (13)$$

$$q_j X_{ijk} \leq vl_{ij} \leq X_{ijk}(Q^k - q_i) : vl_{ij} > 0, \forall k = \{1,2, \dots K\}, \ , \forall i = 1, 2 \dots n, \forall j = 1, 2 \dots n \quad (14)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{K} X_{ijk} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{K} X_{jik} \ , \forall k = \{1,2, \dots K\}, \ i \neq j \quad (15)$$

$$t_{ij} = \frac{d_{ij}}{ADS}, ADS \leq 48 \left[\frac{km}{h}\right], \text{ADS} = \text{Average Drive Speed} \quad (16)$$

$$u_c = (\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{K} d_{ij} \,.X_{ijK})/2 \quad (17)$$

$$X_{ijk} \in \{0,1\} \quad (18)$$

$$u_k \in \{0,1\} \quad (19)$$

$$Y_{jk} \in \{0,1\} \quad (20)$$

Constraint in Eq. 7 implies that the vehicle k^{th} will begin its tour from the depot, while constraint in Eq. 8 ensures the vehicle k^{th} starts picking up the waste from the depot without load. On the other hand, the constraint in Eq. 9 guarantees that when the vehicle visits the last waste container, it must reach the depot [31]. Meanwhile, the constraint in Eq. 10 indicates that a vehicle must fully empty all containers it visits, which is associated with the decision variable Y_{jk}, as expressed in Eq. 3. Eq. 11 presents that the total amount of waste in a route cannot exceed its threshold of vehicle capacity, which is different from the strategy of using fixed vehicle capacity that was confirmed by Lee et al. [32] and Altabeeb et al. [33]. The constraint in Eq. 12 shows that the distance of two nodes travelling back and forth is the same.

Furthermore, the constraint in Eq. 13 ensures that all vehicles are used if they are all maintained, while the constraint in Eq. 14 ensures the vehicle's load (vl_{ij}) is non-negative. Meanwhile, the constraint in Eq. 15 ensures that the vehicle must exit the container (i) after entering it. Constraint in Eq. 16 determines the actual time of the vehicle's movement between containers, including the total movement time, assuming a fixed vehicle's speed is 48 kilometers per hour (km/h). The constraint in Eq. 17 assumes that the cost of used vehicles is equivalent to 50% of the distance cost, while constraints in Equations (18- 20) define the domain of the decision variable.

3.2 Hybridizing of ACO algorithm

ACO is a nature-inspired metaheuristic algorithm [34]. ACO is a swarm algorithm derived from the social behavior of ant colonies that researchers utilize to solve optimization problems [35]. When ants obtain food, they return to their nest through the shortest route possible. During their travels, ants drop pheromones on the ground to indicate which pathways other colony members should follow. Although ACO is robust and can explore a solution, the convergence is still slow and down into the local optimum [36]. Therefore, hybridization is required to improve ACO performance. The whole waste collection system consists of four sections: 1) Construction of the initial solution model; 2) Elimination of a sub-tour problem; 3) Demolition and reconstruction of the solution model, and 4) Selection of optimal solution model, which in turn highlights the contribution explained in the form of steps which will be illustrated in the coming sections (See Fig. 1).

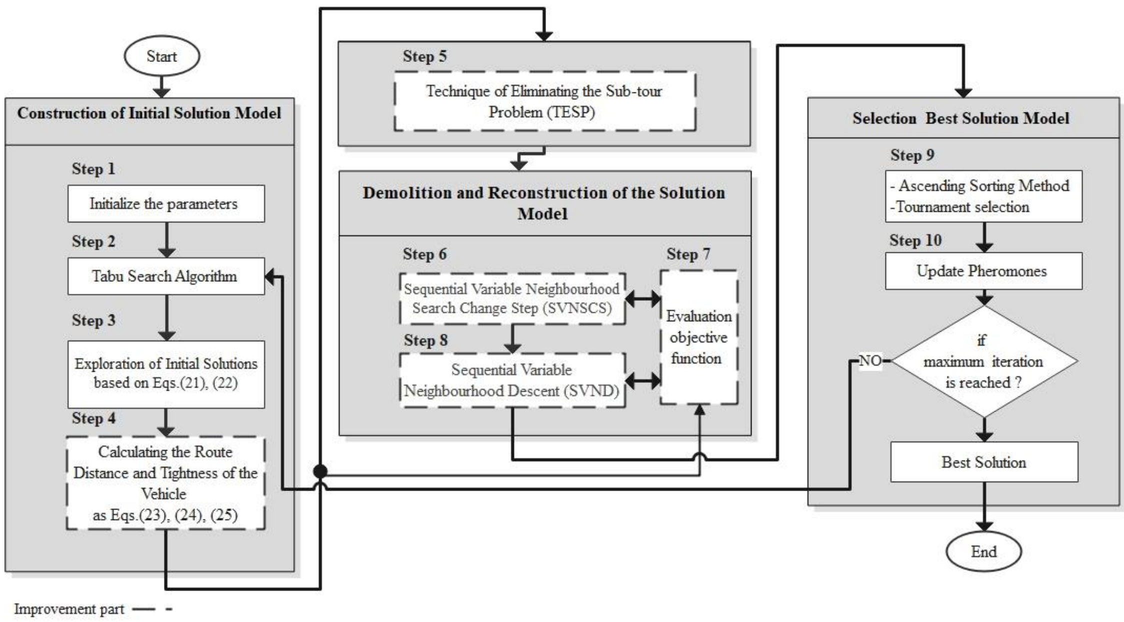

Figure 1. Flowchart of hybridized ACO algorithm

3.3 Construction of initial solution model

The ACO algorithm consists of two fundamental phases: designing the ants' routes and updating pheromones. In a network of n containers, in addition to the location of the depot node, the vehicles simultaneously establish routes starting at initial nodes that are randomly selected. At each building stage, while k is at node (i), a probabilistic random proportional rule determines the next node. This stage demonstrates how it chooses the action to extend its route. For further insight, the ACO will be described in the following five steps:

Step 1: Initialize the parameters

The number of ants, the maximum number of iterations, parameter controls like α and β, and the threshold of demands are set. For the sensitive response of the proposed algorithm in the exploitation stage, the evaporation value is set between [0,1].

Step 2: Tabu search algorithm

In this step, the Tabu search representing local search algorithms may prevent a fall in the optimal local solution [37]. The Tabu list is used initially to cluster the routes and remove the candidate container from the list that has not been serviced. It is a mechanism applied to prevent a vehicle from servicing a container more than once, as outlined in Equation 10. Instead, the search will pick a container and terminate based on the waste collected without exceeding the threshold of the vehicle's capacity, as formulated in Eq. 11. This technique is repeated until all other containers in the route are serviced.

Step 3: Exploration of initial solutions

The initial solution to the vehicle routing problem is a collection of routes with a fixed number of containers. Each solution in the ACO algorithm is constructed using the Tabu search technique and the probability distribution. It is important to note that each route's containers are allocated based on the beginning container that was randomly selected [38], as expressed in Eq. 21.

$$P_{i,j}^{k} = \begin{cases} \frac{(I_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{i \in N_i^k}(I_{ij})^{\alpha}(\eta_{ij})^{\beta}}, & \text{if } j \in N_i^k \\ 0 & \text{, otherwise} \end{cases} \text{(Initial exploration stage)} \tag{21}$$

All operations between container (i) and container (j) in a single edge can be defined as follows.

$P_{i,j}^{k}$: the probability of moving the vehicle in the remaining edges (i, j).
I_{ij}: the intensity of the pheromone.
N_i^k: a set of containers that vehicle (k) has not visited yet.
α: the parameter to regulate the effects of I_{ij}.
β: the parameter to regulate the effects of η_{ij}.
η_{ij}: the inverse of the distance between containers (i) and (j), in other words, represents the shortest distance in edge (i, j), as shown in Eq. 22 [39].

$$\eta_{ij} = 1/\text{Distance between container } (i) \text{ and container } (j) \tag{22}$$

Step 4: Calculating the route distance and tightness of the vehicle

Typically, the total distance for each route is calculated based on the distance matrix d_{ij} along with the initial pheromone matrix, representing crucial variables for determining the optimal route.

Here, the distance calculation between container (i) and container (j) in a 2D space matrix is implemented based on the Euclidean equation shown in Eq. 23.

$$|d_{ij}| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{23}$$

where,

x: location of container source.
y: location of container target.

In this paper, the tightness factor (t_{kho}) is a vital variable that can be determined by dividing the waste volume in each container by the vehicle capacity. It is important to point out that the variable t_{kho} is critical for determining the remaining space in the homogenous vehicle's capacity after merging the sub-tour with the least amount of waste, as expressed in Eq. 24.

$$Total\ q_r = (\textstyle\sum_{j=1}^{n} q_j^k), j \in \{1, \ldots n\},\ k = \{1, \ldots .K\} \tag{24}$$

$Total\ q_r$: total amount of waste for each route.

Let's substitute the value of $Total\ q_r$ in Eq. 25.

$$t_{kho} = \frac{1}{Q_{kho}} \times Total\ q_r,\ j \in \{1, \ldots n\},\ k = \{1, \ldots .K\} \tag{25}$$

Let,

t_{kho}: homogenous tightness variable in the individual route.
Q_{kho} : homogenous vehicles' capacity.

Subsequently, the average tightness in each route can be calculated based on Eq. 26.

$$t_{Akho} = (\textstyle\sum_{j=1}^{n} t_{kho}) / R), j \in \{1, \ldots n\},\ k = \{1, \ldots .K\} \tag{26}$$

t_{Akho}: the average tightness in the solution
R: number of routes

From the Algorithm 1 pseudo-code, a single **for** loop is used to determine the tightness in each container on each route.

Algorithm 1: Pseudo-code of calculating tightness

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

Step 5: Eliminating the sub-tour problem

This research introduces a new technique to reduce the number of vehicles required, thereby reducing the cost of waste transportation by eliminating redundant sub-tours. It also studies the influence of the technique, as shown in the flowchart in Fig. 2.

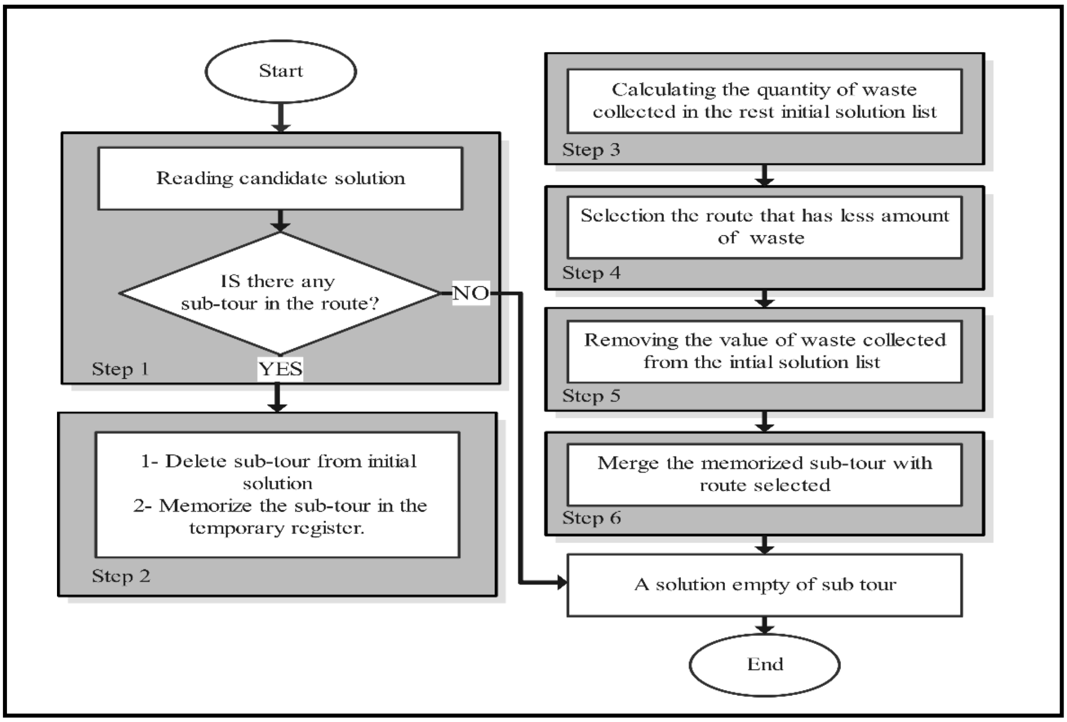

Figure 2. Flowchart of eliminating the sub-tour problem

This technique is considered the primary key to the enhancement of heuristic algorithms. It comprises the following phases: Step 1 will read each route individually and determine if it contains sub-routes. In Step 2, the redundant routes from the current solution will be deleted and saved in a temporary registration. Step 3 calculates the amount of waste for each route. Step 4 determines the route with the least amount of waste. In Step 5, the value of collected waste will be removed from the list of original solutions. Subsequently, integrating the selected sub-tour with another route will reduce waste in Step 6. Eventually, the new route will be reconstructed.

3.4 Demolition and reconstruction of the solution model

Fig. 3 depicts the methodology of hybridizing ACO with a nested metaheuristic algorithm consisting of two phases. The first phase hybridizes ACO with SVNSCS procedures after reading the routes' dynamics and calculating each route's distance. Variable Neighborhood Search (VNS) is a well-known metaheuristic technique to explore the solution space. Specifically, SVNSCS determines which neighborhood will be explored next and whether it will be considered a new solution.

Step 6: Sequential variable neighborhood search change step

This step represents the first phase for hybridizing ACO with SVNSCS procedures after reading the dynamic routes in several iterations and calculating the distance of each route. Feasible solutions have been evaluated in each iteration in the neighborhood change step as a condition cleared in Step 6. The best solution is memorized in the register of the reconstruction solution. Implementing SVNSCS involves the sequential swapping technique on each route as a systematic framework. Other than that, the ratio of best swapping depends on the shortest route to waste CPs, provided that the total waste in a sequence of containers should be less than the vehicle's capacity. Note that the best swapping refers to the best solution. It is selected by evaluating each swapping as Step 8 in Fig. 3.

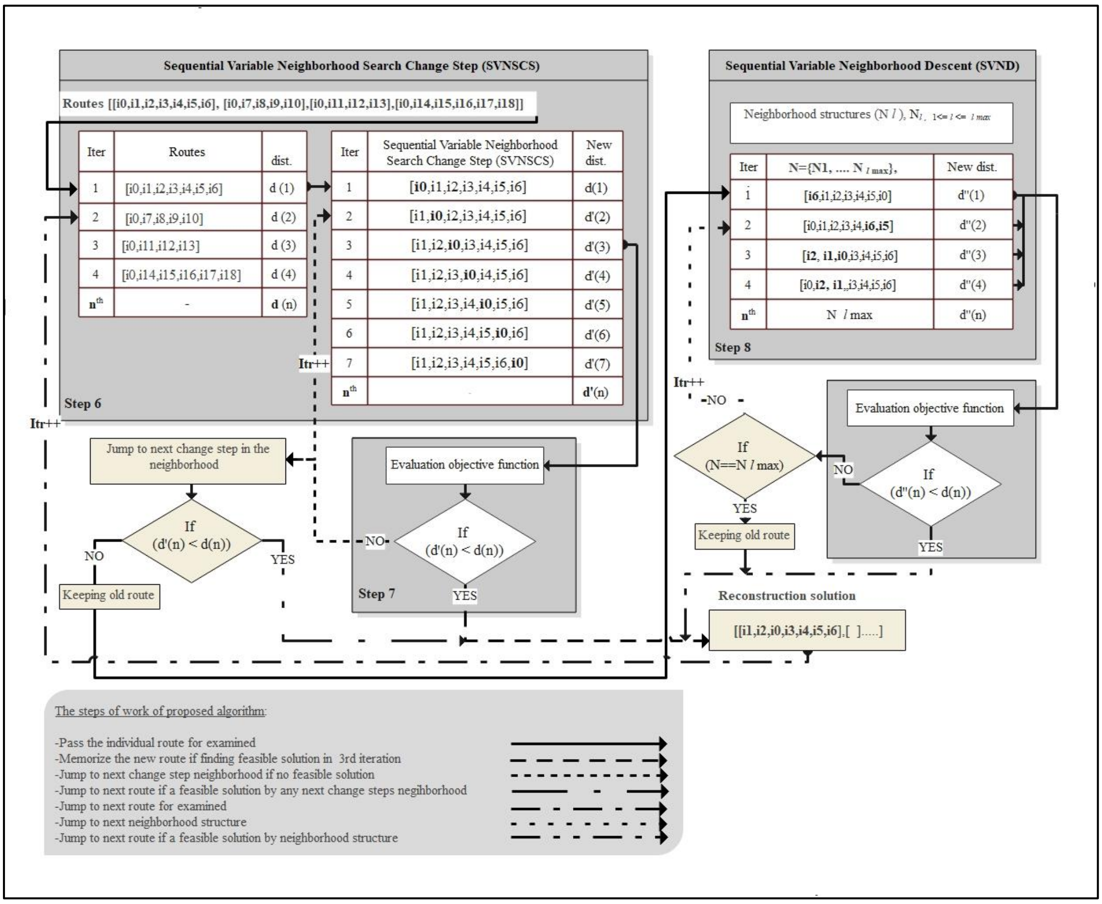

Figure 3. Methodology of a hybridized ACO with SVNSCS and SVND

Step 7: Evaluation of objective function

In this step, the distance between the location of a container and other containers is calculated using probability theory referred to in the initial solution generation model section in Step 4. The new objective function is calculated and compared simultaneously with the previous one.

Step 8: Sequential variable neighborhood descent

Considering the second stage in the demolition and reconstruction solution model, this step begins when the existing route has not been improved in the incumbent route. In this case, it will explore the distant neighborhoods of the incumbent solution and move from there to a new one if the improvement is made. This algorithm is called Sequential Variable Neighborhood Descent (SVND). The SVND method employs a list of ordered neighborhood structures and tests each structure by evaluating their sequence. The SVND principle works as follows: Let $N = \{N1, \ldots, Nl \ max\}$ be a set of operators specifying the neighborhood structures and their testing order. Nevertheless, the fundamental sequential VND process investigates its neighborhood structures defined by the operators $Nl, 1 \leq \ l \ \leq \ l \, max$ sequentially and iteratively. As soon as an improvement to the incumbent solution in a neighborhood structure happens, the basic sequential VND restarts its search in the new incumbent solution's first neighborhood structure (in the defined sequence).

The search continues until the maximum iteration number, returning to the old route and saving the results in the reconstruction solution register. This process applies to any of the *l* max neighborhood structures, as the search operation initiates an evaluation between the various neighborhood structures that explore the best solutions. Since the set of neighborhood operators consists of four structures, the symbol list comprises Per2Opt, Las2Opt, One2Opt2, and Swap1T3, where the first structure exchanges two containers 2Opt at the route list's peripheral. The second structure swaps the route list's final two containers. The third structure swaps index 2 with index 1 on the route list, and the fourth is index 1 with index 2.

All these structures are associated with the symbols mentioned respectively. Note that two motivations exist for utilizing a neighborhood structure above to address the following issues. The first issue is the weaknesses in the ACO algorithm when the ACO algorithm randomly chooses the first node from the list of nodes during the construction of the initial solution. The second issue is that the ACO chooses the optimal route from the last container back to the same depot. The solutions for each change in the neighborhood's structure have been reviewed, verifying their viability. Therefore, switching different neighborhood structures expands the variety of solutions, speeding up the algorithm's convergence.

3.5 Selection of the best solution model

Step 9: Ascending order and tournament selection methods

Typically, ascending order is used to choose the optimal local solution for each iteration in terms of minimum value. The step passes the best local solution to the next generation using a tournament selection technique until it reaches a global solution [40].

Step 10: Update of pheromones

Based on a study proposed by Guo et al. [38], generating the solutions for each iteration depends on the pheromones updated that, in turn, aims to increase the concentration on the regions containing high-quality solutions, as illustrated in Eq. 27.

$$\tau_{ij}^{new} = \rho \times \tau_{ij}^{old} + \sum_{k \in K} \Delta\tau_{ij}^{k} \tag{27}$$

where,

τ_{ij}^{new}: A new deposit of pheromones.
τ_{ij}^{old}: The initial pheromone.
ρ: The evaporation value of pheromone $\rho \in (0,1)$.
k: The vehicle served the routes needed.
K: The number of routes.
$\Delta\tau_{ij}^{(k)}$: The increased pheromone in the edge (i, j).

Eventually, the improved algorithm can be implemented as a waste collection system by specifying the coordinates of collection points (containers) with their waste amounts on a Google map and then converting all points from geographically coordinated to metric coordinates while further replacing the new inputs with information of data algorithm sources.

3.6 Dynamic capacitated vehicles routing model

This model focuses on the methodology of detecting the route that has less tightness to assign the appropriate vehicle. Thus, the model is defined as appropriate vehicle capacity based on the routes tightness model (AVCRTM). Investigating the possibility of achieving this approach can be described as follows.

3.6.1 Heterogeneous vehicles based on route tightness model

Initially, the calculation entails dividing the total waste demands in a specific route over the vehicle capacity. In CVRP, weight refers to the waste demands inside a set of nodes distributed geographically in a two-dimensional area. The HACO algorithm first calculated the tightness of each route and then optimized the routes with lower tightness. It is necessary to calculate threshold levels of homogenous vehicle capacity (T_{kho}) which is by multiplying the Vehicle Capacity (Q_{kho}) with the expected demands in a single route (W) divided by 100, as shown in Eq. 28.

$$T_{kho} = Q_{kho}\left(\frac{W}{100}\right),\ W \in \{80, \mathbf{85}, 90, 95, 100\} \tag{28}$$

Let,

T_{kho}: threshold levels of homogenous vehicle capacity.
Q_{kho}: homogenous vehicle capacity.
W: expected waste demands in an individual route.

Eq. 28 involves multiplying the vehicle capacity by the percentage of waste vehicles permitted to deliver. Researchers can measure the influence of load association with other factors such as fuel consumption, carbon dioxide (CO_2) emission, and more with a novel method for determining the vehicle's capacity. Installation of a microcontroller with a weight sensor in each vehicle is needed to detect the amount of waste inside it. In this case, the vehicle will return to the depot after arriving at the vehicle capacity to the threshold level defined by the waste collection system.

$\{\text{e.g., if } Q_{kho} = 220 \text{ units, and } W = \mathbf{85}\}$

$$T_{kho} = 220\left(\frac{\mathbf{85}}{100}\right)$$
$$T_{kho} = 187$$

The objective of designing the AVCRTM goes in two forms:

1- Restricting the amount of waste allowed to be carried by homogeneous vehicles (Applied dynamic of CVRP).
2- Detection of the route in which the total amount of waste for each route ($Total\ q_r$) less than or equal to threshold levels of Homogenous Vehicle Capacity (T_{kho}). Consequently, assigning appropriate vehicle capacity (heterogeneous vehicle).

The methodology for the second objective of the solid waste collection management system is to maximize the amount of waste collected by vehicles. Firstly, assume a proposed algorithm constructed 7 routes; the sampling methodology of design AVCRTM consists of homogenous vehicle capacity (Q_{kho}), expected waste demands in an individual route (W), reading the routes in each solution, calculate the total amount of waste for each route ($Total\ q_r$), threshold levels of homogenous vehicle capacity (T_{kho}) and current tightness value in the individual route (t_{kho}) as illustrated in Fig. 4.

The bold font in the 3rd iteration indicates the route with a lower tightness value, so the HACO filtered the 3rd route according to the specified condition. Thus, the appropriate vehicle capacity can be calculated using the following mathematical procedures. The difference between the maximum value of tightness ($t_o = 1.0$), and the homogenous tightness (t_{kho}) as expressed in Eq. 29.

$$\Delta_{tg} = t_o - t_{kho} \tag{29}$$

$\Delta_{tg} = 1 - t_{kho}$. According to Figure 4, substitute 0.4 in t_{kho} with reference to Eq. 29, $\Delta_{tg} = 0.6$. Let, Δ_{tg}: tightness generated. For determining the heterogeneous vehicle capacity (Q_{khe}), follow Eq. 30.

$$Q_{khe} = Q_{kho} \times \Delta_{tg} \tag{30}$$

$$Q_{khe} = 220 \times 0.6 = 132$$

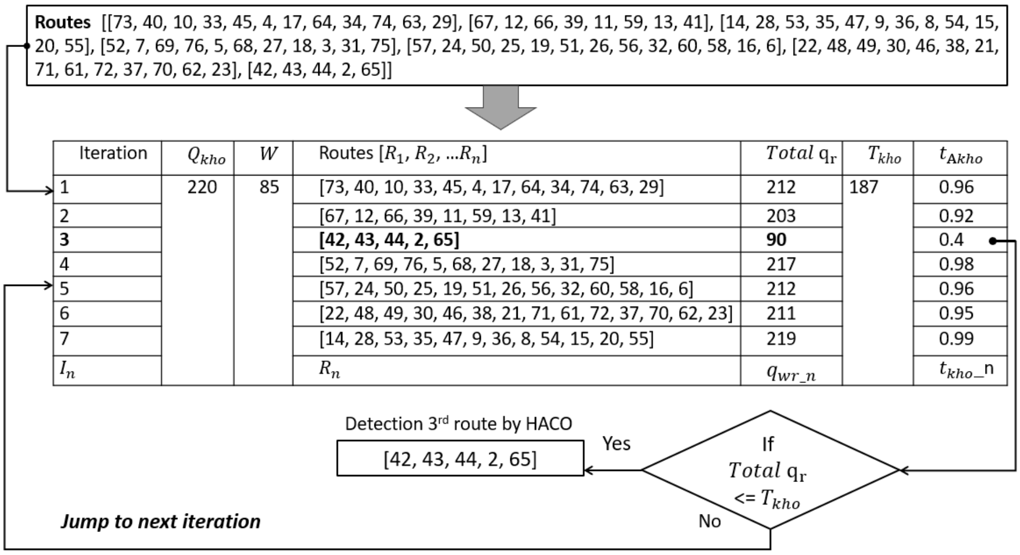

Figure 4. Framework for detecting a route based on lower tightness value

Finding the heterogeneous vehicle capacity component concerning the threshold levels of vehicle capacity (T_{khoe}) based on Eq. 31.

$$T_{khoe} = \frac{Q_{khe} \times W}{100} \tag{31}$$

T_{khe}: threshold levels of heterogeneous vehicle capacity.

$$T_{khoe} = \frac{132 \times 85}{100} = 112.2$$

For testing the acceptability of the heterogenous tightness (t_{khe}), substitute the value of Q_{khe} instead of the value of Q_{kho} that was mentioned in Eq.25. The re-formulation is shown in Eq. 32.

$$t_{khe} = \frac{1}{Q_{khe}} \times Total\ q_r\ ,\ \ j\ \in \{1, \ldots n\},\ \ k = \{1, \ldots . K\} \tag{32}$$

t_{khe}: heterogenous tightness variable in the individual route.

$$t_{khe} = \frac{1}{132} \times 90 = 0.68$$

Therefore, the appropriate heterogeneous vehicle capacity is 132 units because it is located within an acceptable range (0.6-1.0).

4. Results and discussion

The proposed algorithms are evaluated utilizing four benchmark datasets: Dataset A, B, E, and P. Dataset A, B and P were originally from Augerat et al. [41], whereas set E was established by Christofides and Eilon [42]. The instances in this dataset represent different scenarios of the CVRP, where vehicles must deliver goods to various locations while respecting capacity constraints. Each instance is denoted by a combination of parameters, such as the number of nodes (customers), vehicles, and vehicle capacities. The same dataset was also used by Matthopoulos and Sofianopoulou to solve the CVRP problem using the firefly algorithm [43].

In this study, the algorithms are executed ten times in each instance, where each run is terminated when the maximum number of iterations is reached. Moreover, the following parameters of the proposed method are determined based on trial and error: Maximum number of iterations, number of ants, α, β, ρ and τ are 250, 50, 2, 4, 0.7, and 80, respectively. All strategies were coded in

Python and implemented on a PC with 4 GB of RAM and an Intel core i5 processor running at 2.3 GHz.

The format X-nXX-kX is the instance's name, where the variable X is a benchmark name. Besides, nXX indicates the CPs to be serviced, while kX refers to the number of vehicles. The proposed algorithm has been evaluated by measuring the accuracy and comparing the best distance value in ACO and HACO with the BKS of the dataset class, as shown in Equations 33 and 34 [44].

$$\Delta_{d_{ACO}}(\%) = \frac{(d_{ACO}-d_{BKS})}{d_{BKS}} \times 100 \tag{33}$$

$$\Delta_{d_{HACO}}(\%) = \frac{(d_{HACO}-d_{BKS})}{d_{BKS}} \times 100 \tag{34}$$

In addition, the improvement value between the best distance of ACO and the best distance of HACO can be calculated by Eq. 35.

$$\Delta = d_{ACO} - d_{HACO} \tag{35}$$

where,

Δ: the improvement value.

d_{ACO}: the best distance by basic ACO.

d_{HACO}: the best distance by hybridizing ACO.

In addition, the quality of the proposed algorithm was determined by calculating the sum of the best distances (Avg.) optimized and divided by the number of instances in each benchmark class, as shown in Eq. 36.

$$\text{Avg.}(\%) = \left(\frac{1}{n}\sum_{i=1}^{n} d_{HACO}^{i}\right) \tag{36}$$

where,

n: the number of instances.

i: the index of summation best distances.

d_{HACO}^{i}: the distance value for a given index.

4.1 Evaluation of HACO with standard ACO

In this subsection, a comparison is made between standard ACO and HACO in different datasets for the capacitated vehicle routing problem gathered from various sources. The CVRP dataset consists of four different classes. Each has different characteristics regarding the number of nodes, vehicle capacity, objective function, weight of each node, and geographical distribution. The input parameters are the instance column, the problem dimension column (D), the number of vehicles column (K), and the distance of the BKS column (d_{BKS}). The objective function is to reduce the route's distance.

Table 2 displayed the benchmark class A, showing that the HACO outperforms the basic ACO algorithm in 8 instances, as the numbers highlighted in the shaded column, except for A-n32-k5, which was extremely near the optimal solution of the basic ACO algorithm. The proposed algorithm obtained an accuracy of 100% for the best distance. It is important to note that the average values of the suggested method are stable since they are very near to the average value of the best solution. Concerning the tightness value, we observed that all values are close to one, indicating that the proposed method efficiently uses the vehicle's carrying capacity, as shown in the average column.

Table 2. Comparison of HACO with ACO for Benchmark A

				ACO			HACO				
Instances	**D**	**K**	d_{BKS}	d_{ACO}	**Avg.**	$\Delta_{d_{ACO}}$	d_{HACO}	**Avg.**	$\Delta_{d_{HACO}}$	**Δ**	**Avg** $\frac{k}{T}$
A-n32-k5	32	5	784	846	846.3	7.9	788	788.4	0.51	58	1
A-n33-k5	33	5	661	681	681.2	3.0	**678**	681.9	2.57	3	0.8
A-n33-k6	33	6	742	761	762.7	2.6	**756**	756.9	1.89	5.2	0.9
A-n39-k5	39	5	822	882	884.2	7.3	**869**	884.8	5.72	13	0.95
A-n44-k6	44	6	937	975	984.1	4.1	**971**	972.5	3.63	4	0.95
A-n46-k7	46	7	914	1014	1025	10.9	**980**	1045.1	7.22	34	0.86
A-n54-k7	54	7	1167	1256	1256.4	7.6	**1226**	1229.8	5.06	30	0.95
A-n63-k9	63	9	1616	1756	1777.1	8.7	**1707**	1715.1	5.63	49	0.97
A-n69-k9	69	9	1159	1264	1279.5	9.1	**1244**	1245.1	7.33	20	0.93
Avg.							100%				

D: Dimension of a problem; K: Number of vehicles; BKS: best-known solution; Δ: Improvement value; Avg $\frac{k}{T}$: The average of the tightness; HACO-hybridization of ant colony optimization;

Table 3 demonstrates the benchmark class B. It also reveals that the HACO algorithm outperforms the basic ACO algorithm in 11 of 12 instances, most of which are close to the BKS. Except for instances B-n31-k5 and B-n44-k7, they obtain the BKS and surpass it by reducing the number of routes by one, as indicated by the asterisk symbol. It results in an achieved rate of 91.6% for the entire dataset. Each instance's value was extremely near to one regarding the average tightness.

Table 3. Comparison of HACO with ACO for Benchmark B

				ACO			HACO				
Instances	**D**	**K**	d_{BKS}	d_{ACO}	**Avg.**	$\Delta_{d_{ACO}}$	d_{HACO}	**Avg.**	$\Delta_{d_{HACO}}$	**Δ**	**Avg** $\frac{k}{T}$
B-n31-k5	31	5	672	679	680.4	1.0	**595***	596.3	-11.4	84	1.0
B-n34-k5	34	5	788	805	812.1	2.2	**798**	798.8	1.2	7	0.91
B-n38-k6	38	6	805	843	836.8	4.7	**826**	826.6	2.6	17	0.85
B-n45-k5	45	5	751	788	790.1	4.9	**779**	779.6	3.7	9	0.97
B-n43-k6	43	6	742	755	761	1.8	**754**	754.5	1.6	1	0.86
B-n44-k7	44	7	909	967	979.5	6.4	**853***	854.2	-6.1	114	1.0
B-n51-k7	51	7	1032	1085	1089.1	5.1	**1044**	1044.3	1.1	41	0.97
B-n50-k7	50	7	741	798	805.0	7.7	**789**	790.3	6.4	9	0.87
B-n52-k7	52	7	747	784	787.3	4.3	784	786	4.9	0	0.86
B-n56-k7	56	7	707	768	775.2	8.6	**759**	763.1	7.3	9	0.88
B-n66-k9	66	9	1316	1422	1431.3	8.1	**1371**	1376.3	4.1	51	0.95
B-n78-k10	78	10	1221	1331.5	1343.7	9	**1294**	1296.7	5.9	37.5	0.93
Avg.							91.6%				

* Reduce the number of routes by one route

Table 4. Comparison of HACO with ACO for Benchmark E

				ACO			HACO				
Instances	**D**	**K**	d_{BKS}	d_{ACO}	**Avg.**	$\Delta_{d_{ACO}}$	d_{HACO}	**Avg.**	$\Delta_{d_{HACO}}$	Δ	**Avg** $\frac{k}{T}$
E-n22-k4	22	4	375	401	401.5	6.9	**376**	376.5	0.27	25	0.93
E-n33-k4	33	4	835	874	881	4.7	**859**	859.6	2.87	15	0.91
E-n51-k5	51	5	521	611	624.3	17.3	**581**$^{+}$	591.2	11.52	30	0.8
E-n76-k7	76	7	682	797	819	16.9	**775**	776.0	13.64	22	0.88
E-n76-k8	76	8	735	855	863.1	16.3	**823**	826.3	11.97	32	0.94
E-n76-k10	76	10	830	931	958.2	12.2	**913**	918.	10.00	18	0.97
E-n76-k14	76	14	1021	1119	1133.4	9.6	**1098**$^{+}$	1103.1	7.54	21	0.9
E-n101-k8	101	8	817	1000	1017.6	22.4	**964**	971.3	17.99	36	0.91
Avg.							100%				
+ Increase the number of routes by 1											

Table 4 presents the class of benchmark E. Although the HACO is superior to the ACO in all instances with a 100% improvement rate, the instances E-n51-k5 and E-n76-k14 have a slight weakness in increasing the number of routes by one route in comparison to the BKS, where the average tightness is excellent in the proposed algorithm. The proposed algorithm struggled to meet the BKS in all instances except E-n22-k4, resulting in a small size problem. Consequently, the improvement rate of the HACO algorithm in comparison to the ACO was relatively high in all instances. The average of the best solution in the proposed algorithm demonstrates the algorithm's stability.

Table 5 tabulates the results from benchmark P. The HACO significantly outperformed ACO by 87%, except in one instance, P-n45-k5, which obtained the same best distance as ACO. In another instance, P-n70-k10 suffered from an increased number of routes compared to the BKS. Therefore, the tightness value indicates good exploitation in the vehicle capacity.

Table 5. Comparison of HACO with ACO for Benchmark P

				ACO			HACO				
Instances	**D**	**K**	d_{BKS}	d_{ACO}	**Avg.**	$\Delta_{d_{ACO}}$	d_{HACO}	**Avg.**	$\Delta_{d_{HACO}}$	Δ	**Avg** $\frac{k}{T}$
P-n60-k10	60	10	744	830.2	838.4	11.59	**795**	801.3	6.85	35.2	0.94
P-n60-k15	60	15	968	1042	1056.1	7.64	**1009**	1013	4.24	33	0.94
P-n45-k5	45	5	510	577	587.3	13.14	577	579.3	10.20	15	0.92
P-n65-k10	65	10	792	889	908.9	12.25	**859**	861.9	8.46	30	0.93
P-n76-k4	76	4	593	690	701.7	16.36	**665**	666.7	12.14	25	0.97
P-n70-k10	70	10	827	931.6	949.4	12.65	**900**$^{+}$	905.9	8.83	31.6	0.88
P-n76-k5	76	5	627	733	754.9	16.91	**709**	712.9	13.08	24	0.97
P-n101-k4	101	4	681	821	840.1	20.56	**804**	815.8	18.06	17	0.91
Avg.							87%				
+ Increase the number of routes by 1											

4.2 Analysis of multiple improvements of ACO

This section compares basic ACO, ACO-SVNSCS, and HACO based on their optimal distance and run time. The HACO algorithm combines the hybridized ACO with the SVNSCS and VNSD algorithms to assess the effects of modification strategies. We also measured the individual percentage impact of contributing ACO algorithms. Based on the simulation environment, all algorithms are conducted individually with a specified number of iterations and the number of ants: 100 iterations and 25 ants, respectively. However, 8 instances were chosen to diversify the test and ensure the results' accuracy. The instance was picked based on several criteria from various benchmark classes, including vehicle capacity, container count, and route count.

The results of HACO were superior to ACO in 7 out of 8 instances, and they achieved a score of 87.5%. Although the run time was close to the run time in the rest of the ACO, the improvement rate was excellent. However, in only one case, in the instance P-n19-k2, there was no difference compared to other algorithms. Therefore, it was concluded that the closer the distribution of container locations is to each other, the better the impact of the optimization techniques. Accordingly, computational results prove the effectiveness and efficacy of the strategy proposed, as shown in Table 6.

Table 6. Results of comparison impact for different ACO

		ACO			ACO-SVNSCS			HACO		
Instances	**Q**	**d_{ACO}**	**Avg.**	**Time (sec)**	**$d_{ACO-SVNSCS}$**	**Avg.**	**Time (sec)**	**d_{HACO}**	**Avg.**	**Time (sec)**
A-n33-k5	100	703	705.6	51	701	717	57	**692**	706.28	61.8
A-n60-k9	100	1546	1575.4	115.2	1523	1550	119.4	**1516**	1516	120
B-n31-k5	100	685	687.5	47.4	610	601.6	52.2	**595**	598.91	52.8
B-n50-k7	100	807	812.4	85.8	806	812	94.2	**805**	810.57	94.2
E-n22-k4	6000	433	437.5	33	426	438.8	32.4	**391**	413.04	34.8
P-n19-k2	160	218	218.5	27	221	221.8	29.4	218	218.5	**28.8**
P-n70-k10	135	1017	1022.7	141.6	984	984.34	146.4	**965**	989.3	148.2
P-n101-k4	400	871	875.04	223.9	879	891.4	232.02	**863**	869.22	300
Avg.					75%			87.5%		

ACO: Ant Colony Optimization; SVNSCS: Sequential Variable Neighborhood Search Change Step; HACO: Hybrid ACO with (SVNSCS+SVND); Q: Vehicle capacity

The ACO-SVNSCS outperforms the ACO in 6 out of 8 instances at a rate of 75%. In terms of best distance, the results were quite similar. Despite the adjustment of SVNSCS, the run time values were comparable to those of ACO, although it outperformed the run time in the instance of E-n22-k4, distinguished by its vast capacity. HACO was superior to ACO in 7 out of 8 instances and earned an overall score of 87.5%. The runtime was comparable to the rest of the ACO, and the rate of improvement was remarkably high.

4.3 Analysis of eliminating sub-tour problem

This section will clarify the impact of eliminating the sub-route problem on waste transportation costs by comparing three examples. These examples were chosen to diversify the comparison onto a specific geographical distribution of containers. The benefits of this strategy are summarized as follows:

1. It is considered the primary key to improving the algorithm of ant colonies with local search algorithms.
2. Reducing transportation costs by merging the sub-route with the tracks that have the least waste is directly proportional to reducing the number of vehicles used.
3. Reducing the distance travelled to serve the CPs.

Table 7 presents the computation results, showing the difference between the route structure with and without the sub-tour.

Table 7. Index of different instances

Problem	Instance	Problem	Instance	Problem	Instance
P1	A-n33-k5	P2	E-n76-k10	P3	B-n44-k7

Different instances with and without the sub-tour have been compared. The same algorithm's parameters are used, such as assigning 5 as some iterations and some ants in a single run. According to that, the performance of the proposed algorithm gives optimal routes in all instances selected, compared to the previous ACO, as shown in Table 8. This outcome reduced the distance of 12.19, 160, and 28.16 units, respectively, based on Equation 37. Figure 5 demonstrates the number of routes reduced in different instances: A-n33-k5, B-n44-k7, and E-n76-k10.

$$\Delta_S = D_s - D_{ws} \tag{37}$$

where,

Δ_S: improved distance after eliminating the sub-tour.
D_s: distance with sub-tour.
D_{ws}: distance without sub-tour.

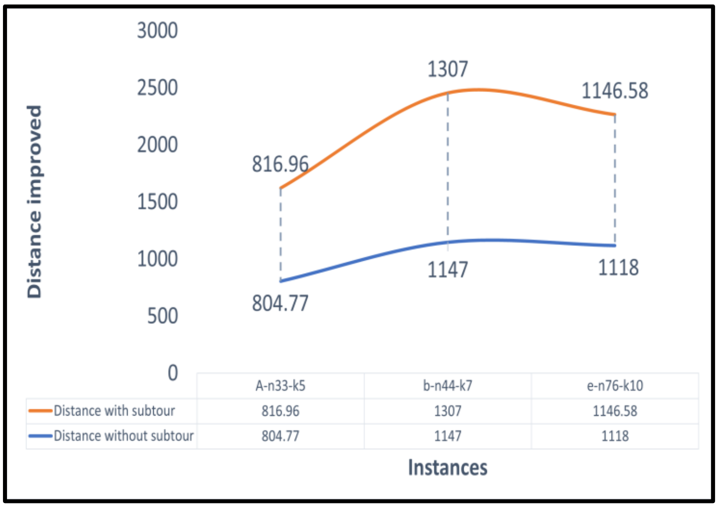

Figure 5. The impacts of eliminating sub-tour

Table 8. Effects of eliminating the sub-tour problem in different instances

Prob.	Routes with sub-tour	N_R	D_s	Routes without sub-tour	$\overline{N}_R$	D_{ws}	Δ_S
P1	[[19, 29, 24, 12, 7, 25], [32, 22, 15, 20, 2, 30, 17, 4], [33, 21, 5, 13, 11, 31, 26], [9, 14, 27, 8, 6, 28], [10, 18, 16, 23], [3]]	6	816.96	[[10, 18, 16, 23], [9, 14, 27, 8, 6, 28], [33, 21, 5, 13, 11, 31, 26], [32, 22, 15, 20, 2, 30, 17, 4], [19, 29, 24, 12, 7, 25, 3]]	5	804. 77	12.19
P2	[[16, 34, 4, 44, 32], [6, 18, 41, 2, 13, 29, 26, 21], [27, 15, 37, 3, 40, 25, 36], [38, 42, 35, 14, 22, 7, 33, 20], [5, 10, 12, 11], [28, 19, 17, 30, 8, 9, 31], [43, 23, 39], [24]]	8	1307	[[43, 23, 39], [27, 15, 37, 3, 40, 25, 36], [16, 34, 4, 44, 32], [38, 42, 35, 14, 22, 7, 33, 20], [6, 18, 41, 2, 13, 29, 26, 21], [28, 19, 17, 30, 8, 9, 31], [5, 10, 12, 11, 24]]	7	1147	160
P3	[[76, 69, 7, 52, 18, 41, 13], [72, 61, 71, 21, 38, 6, 30, 46, 28], [62, 29, 63, 74, 34, 64, 17, 4], [36, 8, 9, 47, 35, 53, 14, 55], [51, 19, 26, 56, 10, 40, 73, 59], [33, 45, 50, 25, 24, 57, 42, 43], [48, 22, 75, 31, 3, 5], [20, 15, 54, 12, 66, 39], [49, 37, 70, 16, 58, 68, 27, 11], [23, 2, 44, 65, 32, 67], [60]]	11	1146.58	[[76, 69, 7, 52, 18, 41, 13], [36, 8, 9, 47, 35, 53, 14, 55], [72, 61, 71, 21, 38, 6, 30, 46, 28], [33, 45, 50, 25, 24, 57, 42, 43], [48, 22, 75, 31, 3, 5], [62, 29, 63, 74, 34, 64, 17, 4], [49, 37, 70, 16, 58, 68, 27, 11], [20, 15, 54, 12, 66, 39], [23, 2, 44, 65, 32, 67], [51, 19, 26, 56, 10, 40, 73, 59, 60]]	10	1118.42	28.16

N_R: Number of routes before improvement; $\overline{N}_R$: Number of routes after improvement;

4.4 Sensitivity analysis of vehicle capacity

Fig. 6 displays the outcomes of the sensitivity analysis. It demonstrates how restricting the search space in terms of vehicle capacity dynamics impacts the optimal route. The sensitivity of HACO is performed to investigate the effects of different percentage values (80-100%), representing the amount of waste allowed to be collected by vehicle capacity in each route. This method enables HACO to limit a precise amount of waste, controlling the vehicle's capacity. The simulation results indicate that a 100% level of waste in the vehicle capacity provides the best distance improvement, based on the first 40 iterations, compared to another percentage of capacity levels. Furthermore, it converges quickly in the 370 iterations because it has a more extensive search area. In contrast, increasing the percentage values for HACO takes longer, as shown in Table 9.

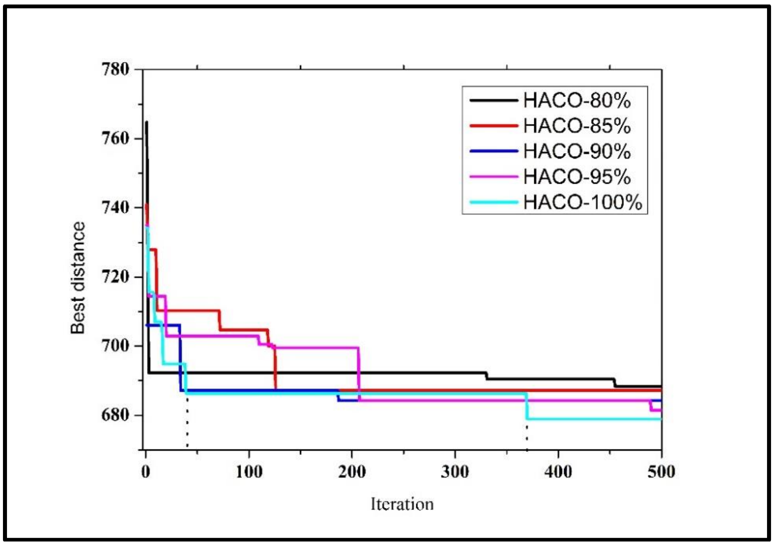

Figure 6. The performance of HACO in the dynamic of vehicle capacity

Table 9. HACO's Vehicle Capacity Analysis

Level of Q (%)	Best distance	Avg.	Run time (sec)
80	688	691.66	91.7
85	687	692.66	94.47
90	684	686.57	101.56
95	681	691.73	101.71
100	**678**	**685.64**	142.23

Concerning the run time differences, we notice an inverse relationship between the improvement rate and the level of waste allowed to be carried in vehicles. As the search area increases, the improvement rate increases. When a level of Q = 100 is chosen, it means an increase in time complexity and vice versa. When a level of Q = 80 is chosen, only the path matching the specified threshold is optimized.

4.5 Convergence behavior analysis

The convergence behavior of ACO, ACO-SVNSCS and HACO algorithms was also analyzed. The algorithm parameters are 500, 25, and 100 for the number of iterations, the number of ants, and the percentage of waste allowed in the vehicle's capacity. To ensure the diversity of the solution, we examine two examples: a) A-n33-k5 and b) P-n101-k4. In instance (a), the initial improvement ACO-SVNSCS converged locally quicker than HACO in the first 50 iterations. This indicates that ACO-SVNSCS progressed rapidly from the existing best solution to the best new solution after 150 iterations. Hence, HACO has more potential to escape from local optima than basic ACO and ACO-SVNSCS. However, the latter converges faster than basic ACO, as seen in Figure 7. In instance (b), it is apparent that HACO has a greater chance of converging within the first 50 iterations than ACO-SVNSCS and basic ACO. In 220 iterations, the second convergence demonstrates that HACO outperforms the other methods, and the improvement of ACO-SVNSCS still outperforms the convergence of basic ACO.

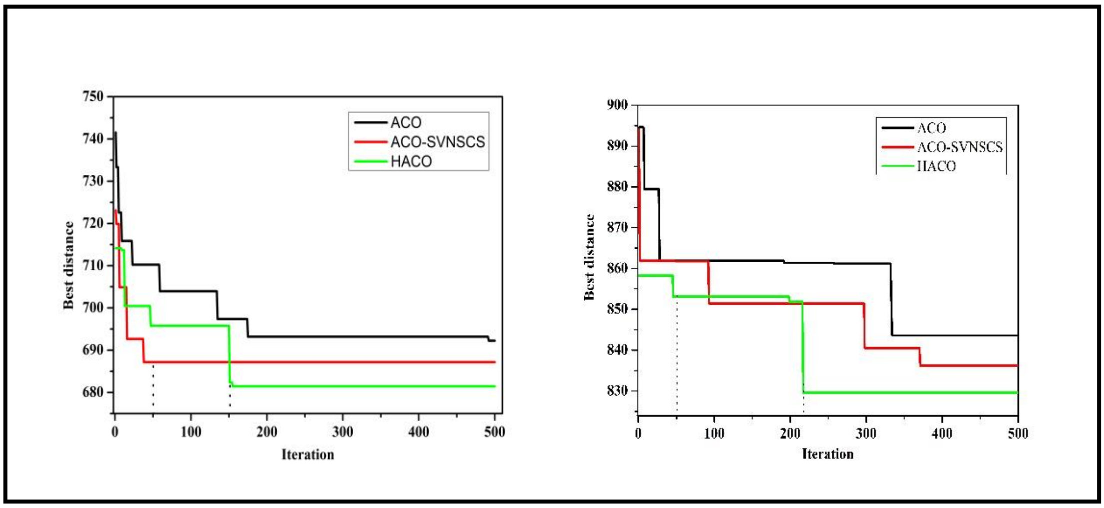

Figure 7. Convergence behavior for instances A-n33-k5 (left) and P-n101-k4 (right)

5. Conclusion

This paper presented a comparative study of the hybrid ACO with nested VNS algorithms, including SVNSCS and SVND, for solving CVRP and DCVRP for waste collection systems. A systematic framework model was designed by hybridizing ACO to increase its searchability in the exploration and exploitation stages. Variants of average vehicle speed and threshold of vehicle capacity were adopted to observe the arrival time and analyze the effect of dynamic vehicle capacity on the VRP. Simulation experiments demonstrated that the proposed method performed significantly better than standard ACO. Four criteria have been used to improve waste collection distance using ACO algorithms. From the results, ACO-SVNSCS outperforms ACO by 75%, and HACO outperforms other ACO algorithms by 85% in minimizing the optimal distance. Furthermore, a new technique was designed to eliminate the sub-tour problem in the ACO algorithms. The computation results demonstrated its superiority in all tests. Moreover, the sensitivity of HACO concerning vehicle capacity was analyzed at five percentage levels. The experimental results concluded that the optimal distance provided at the rate of 100% of the waste amount allowed in the vehicle's capacity. In contrast, it causes an increase in the algorithm's runtime. Exploring the convergence behavior of ACO, ACO-SVNSCS, and HACO revealed that HACO yielded the best results compared to other ACO algorithms. This study only limits the performance of ACO algorithms for solving the dynamic capacity of vehicle routing problems in waste collection systems. Future research may focus on increasing the exploration stage to accelerate the convergence of HACO in large-scale problems, estimating the variant of a daily traffic congestion time, and evaluating the proposed algorithm in the context of other different metaheuristic algorithms for solving the same problems.

Author contribution

T.M. Sahib: Writing - Original Draft, Conceptualization, Investigation, Formal Analysis. R. Mohd-Mokhtar: Supervision, Resources, Writing - Review & Editing. A. Mohd-Kassim: Writing - Review & Editing.

Funding statement

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

Acknowledgements

The authors would like to express their sincere gratitude to the School of Electrical and Electronic Engineering, USM for the support received to conduct this research project.

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Md. M. Hoque and M. T. U. Rahman, "Landfill area estimation based on solid waste collection prediction using ANN model and final waste disposal options," *J Clean Prod*, vol. 256, p. 120387, May 2020, https://doi.org/10.1016/j.jclepro.2020.120387

[2] S. O. Kwakye, E. E. Y. Amuah, K. A. Ankoma, E. B. Agyemang, and B.-G. Owusu, "Understanding the performance and challenges of solid waste management in an emerging megacity: Insights from the developing world," *Environmental Challenges*, vol. 14, p. 100805, Jan. 2024, https://doi.org/10.1016/j.envc.2023.100805

[3] M. Al Duhayyim *et al.*, "Deep Reinforcement Learning Enabled Smart City Recycling Waste Object Classification," *Computers, Materials & Continua*, vol. 71, no. 3, pp. 5699–5715, 2022, https://doi.org/10.32604/cmc.2022.024431

[4] S. Saxena, C. Rajendran, V. Sanjeevi, and P. Shahabudeen, "Optimization of solid waste management in a metropolitan city," *Mater Today Proc*, vol. 46, pp. 8231–8238, 2021, https://doi.org/10.1016/j.matpr.2021.03.219

[5] G. Boskovic, N. Jovicic, S. Jovanovic, and V. Simovic, "Calculating the costs of waste collection: A methodological proposal," *Waste Management & Research: The Journal for a Sustainable Circular Economy*, vol. 34, no. 8, pp. 775–783, Aug. 2016, https://doi.org/10.1177/0734242X16654980

[6] O. Munyai and W. N. Nunu, "Health effects associated with proximity to waste collection points in Beitbridge Municipality, Zimbabwe," *Waste Management*, vol. 105, pp. 501–510, Mar. 2020, https://doi.org/10.1016/j.wasman.2020.02.041

[7] A. S. Silva *et al.*, "Capacitated Waste Collection Problem Solution Using an Open-Source Tool," *Computers*, vol. 12, no. 1, p. 15, Jan. 2023, https://doi.org/10.3390/computers12010015

[8] S. Kaza, L. C. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. 2018. https://doi.org/10.1596/978-1-4648-1329-0

[9] A. Iqbal *et al.*, "Municipal Solid Waste Collection and Haulage Modeling Design for Lahore, Pakistan: Transition toward Sustainability and Circular Economy," *Sustainability*, vol. 14, no. 23, p. 16234, Dec. 2022, https://doi.org/10.3390/su142316234

[10] R. Aziz, G. P. Adfuza, and Y. Arbi, "Preliminary Study of Solid Waste Treatment of Padang Beach Tourism Area, Padang City," *Jurnal Pendidikan Teknologi Kejuruan*, vol. 4, no. 1, pp. 31–34, Jun. 2021, https://doi.org/10.24036/jptk.v4i1.16323

[11] G. Rahmanifar, M. Mohammadi, M. Hajiaghaei-Keshteli, G. Fusco, and C. Colombaroni, "An integrated temporal and spatial synchronization for two-echelon vehicle routing problem in waste collection system," *J Ind Inf Integr*, vol. 40, p. 100611, Jul. 2024, https://doi.org/10.1016/j.jii.2024.100611

[12] N. Rattanawai, S. Arunyanart, and S. Pathumnakul, "Optimizing municipal solid waste collection vehicle routing with a priority on infectious waste in a mountainous city landscape context," *Transp Res Interdiscip Perspect*, vol. 24, p. 101066, Mar. 2024, https://doi.org/10.1016/j.trip.2024.101066

[13] C. Wei, S. Wøhlk, and A. Che, "A multi-level capacitated arc routing problem with intermediate facilities in waste collection," *Comput Oper Res*, vol. 167, p. 106671, Jul. 2024, https://doi.org/10.1016/j.cor.2024.106671

[14] A. Chaabane, J. Montecinos, M. Ouhimmou, and A. Khabou, "Vehicle routing problem for reverse logistics of End-of-Life Vehicles (ELVs)," *Waste Management*, vol. 120, pp. 209–220, Feb. 2021, https://doi.org/10.1016/j.wasman.2020.11.008

[15] T. M. Sahib, R. Mohd Mokhtar, A. F. H. Alharan, N. S. Ali, and A. A. Kadhum, "An Improved Strategy for Solid Waste Management based on Programmable Logic Controller," *Int J Environ Waste Manag*, vol. 1, no. 1, p. 1, 2023, https://doi.org/10.1504/IJEWM.2023.10048103

[16] M. Akhtar, M. A. Hannan, R. A. Begum, H. Basri, and E. Scavino, "Backtracking search algorithm in CVRP models for efficient solid waste collection and route optimization," *Waste Management*, vol. 61, pp. 117–128, Mar. 2017, https://doi.org/10.1016/j.wasman.2017.01.022

[17] B.-I. Kim, S. Kim, and S. Sahoo, "Waste collection vehicle routing problem with time windows," *Comput Oper Res*, vol. 33, no. 12, pp. 3624–3642, Dec. 2006, https://doi.org/10.1016/j.cor.2005.02.045

[18] A. M. Benjamin and J. E. Beasley, "Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities," *Comput Oper Res*, vol. 37, no. 12, pp. 2270–2280, Dec. 2010, https://doi.org/10.1016/j.cor.2010.03.019

[19] T. M. Sahib, R. Mohd-Mokhtar, and A. M. Kassim, "Survey on Meta-Heuristic Algorithms for Solving Vehicle Route Problems in a Waste Collection System," in *LNEE*, vol. 829, 2022, pp. 363–369. https://doi.org/10.1007/978-981-16-8129-5_57

[20] E. B. Tirkolaee, I. Mahdavi, and M. Mehdi Seyyed Esfahani, "A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time," *Waste Management*, vol. 76, pp. 138–146, Jun. 2018, https://doi.org/10.1016/j.wasman.2018.03.015

[21] T. Henke, M. G. Speranza, and G. Wäscher, "The multi-compartment vehicle routing problem with flexible compartment sizes," *Eur J Oper Res*, vol. 246, no. 3, pp. 730–743, Nov. 2015, https://doi.org/10.1016/j.ejor.2015.05.020

[22] Y. P. Li, G. H. Huang, and S. L. Nie, "A mathematical model for identifying an optimal waste management policy under uncertainty," *Appl Math Model*, vol. 36, no. 6, pp. 2658–2673, Jun. 2012, https://doi.org/10.1016/j.apm.2011.09.049

[23] G. Ghiani, A. Manni, E. Manni, and V. Moretto, "Optimizing a waste collection system with solid waste transfer stations," *Comput Ind Eng*, vol. 161, p. 107618, Nov. 2021, https://doi.org/10.1016/j.cie.2021.107618

[24] A. A. Ismail, R. Krisnaputra, and I. Bahiuddin, "Application of Ant Colony Optimization for the Shortest Path Problem of Waste Collection Process," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, Aug. 2021, https://doi.org/10.22219/kinetik.v6i3.1307

[25] J. Edwards, M. Othman, S. Burn, and E. Crossin, "Energy and time modelling of kerbside waste collection: Changes incurred when adding source separated food waste," *Waste Management*, vol. 56, pp. 454–465, Oct. 2016, https://doi.org/10.1016/j.wasman.2016.06.033

[26] F. S. Sahib and N. S. Hadi, "Truck route optimization in Karbala city for solid waste collection," *Mater Today Proc*, vol. 80, pp. 2489–2494, 2023, https://doi.org/10.1016/j.matpr.2021.06.394

[27] J. A. Knowles, "National solid waste management plan for Iraq," *Waste Management & Research: The Journal for a Sustainable Circular Economy*, vol. 27, no. 4, pp. 322–327, Jun. 2009, https://doi.org/10.1177/0734242X09104129

[28] N. A. Mat, A. M. Benjamin, and S. Abdul-Rahman, "Enhanced heuristic algorithms with a vehicle travel speed model for time-dependent vehicle routing: A waste collection problem," *Journal of Information and Communication Technology*, vol. 17, no. 1, 2018, https://doi.org/10.32890/jict2018.17.1.8245

[29] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "An adaptive hybrid algorithm for vehicle routing problems with time windows," *Comput Ind Eng*, vol. 113, pp. 382–391, Nov. 2017, https://doi.org/10.1016/j.cie.2017.09.034

[30] M. A. Hannan, M. Akhtar, R. A. Begum, H. Basri, A. Hussain, and E. Scavino, "Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm," *Waste Management*, vol. 71, pp. 31–41, Jan. 2018, https://doi.org/10.1016/j.wasman.2017.10.019

[31] İ. İLHAN, "An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem," *Swarm Evol Comput*, vol. 64, p. 100911, Jul. 2021, https://doi.org/10.1016/j.swevo.2021.100911

[32] C.-Y. Lee, Z.-J. Lee, S.-W. Lin, and K.-C. Ying, "An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem," *Applied Intelligence*, vol. 32, no. 1, pp. 88–95, Feb. 2010, https://doi.org/10.1007/s10489-008-0136-9

[33] A. M. Altabeeb, A. M. Mohsen, and A. Ghallab, "An improved hybrid firefly algorithm for capacitated vehicle routing problem," *Appl Soft Comput*, vol. 84, p. 105728, Nov. 2019, https://doi.org/10.1016/j.asoc.2019.105728

[34] H. Ali and A. K. Kar, "Discriminant Analysis using Ant Colony Optimization – An Intra-Algorithm Exploration," *Procedia Comput Sci*, vol. 132, pp. 880–889, 2018, https://doi.org/10.1016/j.procs.2018.05.100

[35] M. Pedemonte, S. Nesmachnow, and H. Cancela, "A survey on parallel ant colony optimization," *Appl Soft Comput*, vol. 11, no. 8, pp. 5181–5197, Dec. 2011, https://doi.org/10.1016/j.asoc.2011.05.042

[36] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem," *Eng Appl Artif Intell*, vol. 92, p. 103649, Jun. 2020, https://doi.org/10.1016/j.engappai.2020.103649

[37] S. Faiz, S. Krichen, and W. Inoubli, "A DSS based on GIS and Tabu search for solving the CVRP: The Tunisian case," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 17, no. 1, pp. 105–110, Jun. 2014, https://doi.org/10.1016/j.ejrs.2013.10.001

[38] N. Guo, B. Qian, R. Hu, H. P. Jin, and F. H. Xiang, "A Hybrid Ant Colony Optimization Algorithm for Multi-Compartment Vehicle Routing Problem," *Complexity*, vol. 2020, pp. 1–14, Oct. 2020, https://doi.org/10.1155/2020/8839526

[39] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and G. Paredes-Belmar, "Application of the ant colony optimization in the resolution of the bridge inspection routing problem," *Appl Soft Comput*, vol. 65, pp. 443–461, Apr. 2018, https://doi.org/10.1016/j.asoc.2018.01.034

[40] Z. Borčinová, "Kernel Search for the Capacitated Vehicle Routing Problem," *Applied Sciences*, vol. 12, no. 22, p. 11421, Nov. 2022, https://doi.org/10.3390/app122211421

[41] P. Augerat, D. Naddef, J.-M. Belenguer, E. Benavent, Á. Corberán, and G. Rinaldi, "Computational results with a branch and cut code for the capacitated vehicle routing problem," 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:60747070

[42] N. Christofides and S. Eilon, "An Algorithm for the Vehicle-Dispatching Problem," *OR*, vol. 20, no. 3, 1969, https://doi.org/10.2307/3008733

[43] P. P. Matthopoulos and S. Sofianopoulou, "A firefly algorithm for the heterogeneous fixed fleet vehicle routing problem," *International Journal of Industrial and Systems Engineering*, vol. 33, no. 2, p. 204, 2019, https://doi.org/10.1504/IJISE.2019.102471

[44] F. Arnold, Í. Santana, K. Sörensen, and T. Vidal, "PILS: Exploring high-order neighborhoods by pattern mining and injection," *Pattern Recognit*, vol. 116, p. 107957, Aug. 2021, https://doi.org/10.1016/j.patcog.2021.107957