Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2023, Article ID 5398400, 11 pages
https://doi.org/10.1155/2023/5398400

Research Article

Local Search-Based Metaheuristic Methods for the Solid Waste Collection Problem

Haneen Algethami

Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia

Correspondence should be addressed to Haneen Algethami; hmgethami@tu.edu.sa

Received 29 October 2022; Revised 7 August 2023; Accepted 29 August 2023; Published 6 October 2023

Academic Editor: Francesco Carlo Morabito

The solid waste collection problem refers to truck route optimisation to collect waste from containers across various locations. Recent concerns exist over the impact of solid waste management on the environment. Hence, it is necessary to find feasible routes while minimising operational costs and fuel consumption. In this paper, in order to reduce fuel consumption, the number of trucks used is considered in the objective function along with the waste load and the travelling time. With the current computational capabilities, finding an optimal solution is challenging. Thus, this study aims to investigate the effect of well-known metaheuristic methods on this problem's objective function and computational times. The routing solver in the Google OR-tools solver is utilised with three well-known metaheuristic methods for neighbourhood exploration: a guided local search (GLS), a tabu search (TS), and simulated annealing (SA), with two initialisation strategies, Clarke and Wright's algorithm and the nearest neighbour algorithm. Results showed that optimal solutions are found in faster computational times than using only an IP solver, especially for large instances. Local search methods, notably GLS, have significantly improved the route construction process. The nearest neighbour algorithm has often outperformed the Clarke and Wright's methods. The findings here can be applied to improve operations in Saudi Arabia's waste management sector.

1. Introduction

Solid waste management is a challenging task that is common across cities worldwide [1]. Households generate solid waste, as do commercial and industrial organisations, institutions such as schools, hospitals, care homes, and individuals, and public spaces such as streets, markets, slaughterhouses, public toilets, bus stops, parks, and gardens [2]. This paper considers the solid waste collection problem (SWCP) as managing the waste collection process by loading waste into waste collection trucks from various collection facilities, transporting it, and unloading it at the available disposal sites.

The increasing significance of the management of solid waste is caused by the negative effects of waste buildup on the environment, such as air and water pollution, habitat destruction, and climate change (vergara 2012 municipal). Additionally, it requires considerable costs, as it necessitates the allocation of financial, human, and technical resources for its handling [1]. Poor waste management practices can lead to economic losses through reduced resource recovery, increased healthcare costs due to pollution-related illnesses, and potential damage to tourism industries. Note that the transport sector accounts for around 27% of the overall carbon dioxide emissions [3].

SWCP is critical for cities in urban areas within metropolitan regions, as well as in cities that are responsible for organising significant sporting, entertainment, or religious gatherings [4]. To decrease the carbon footprint of the waste collection process, this problem considers minimising the travelling cost and fuel usage by reducing the waste load while using the lowest number of trucks possible [5]. Therefore, the implementation of an effective waste collection system that minimises fuel usage is essential [6].

Regarding computational complexity, SWCP is considered an extension of the vehicle routing problem (VRP) and the vehicle routing problem with time windows (VRPTW). Both problems are considered NP-hard problems [7]. That is, there is no known polynomial-time algorithm to solve the problem. Therefore, SWCP is considered an NP-hard problem. Even for small instances, finding an optimal solution is hard for commercial solvers [8]. Real-world instances are always a challenge to solve, especially in metropolitan areas. Thus, a heuristic method must be used to obtain at least a feasible solution. This paper aims to investigate well-known metaheuristic methods to generate feasible solutions to the SWCP while considering all conditions and limitations. A guided local search (GLS), a tabu search (TS), and simulated annealing (SA) are all tested with two initialisation strategies: Clarke and Wright's algorithm as well as the nearest neighbour algorithm. The routing solver in the Google OR-tools solver is used. To this end, the specific objectives of this study are as follows:

(i) To investigate the performance of local search-based metaheuristic methods for SWCP that tackle real-world instances from Mecca city.

(ii) To complete illustrative computational trials to establish the utility of the proposed approach.

The subsequent sections of the study are structured as follows. After the introduction, an analysis of the relevant literature is presented. Then, the problem and the instances used in this paper are described. Following that, the methods used are explained. Next, the experimental tests are presented, followed by a discussion of the obtained results. At last, the last part serves to provide a conclusive ending to the article.

2. Related Work

In this section, current and relevant research on SWCP-related works is examined including the constraints, objectives, and metaheuristic application.

Considerable research has been carried out in the field of efficient waste collection, focusing on typical optimisation problems such as VRP and VRPTW, particularly in applying metaheuristic techniques to solve these problems [1, 6, 9]. The interest in developing metaheuristic approaches for difficult combinatorial optimisation problems such as the one tackled in this paper was to reduce the computational cost of these problems. Therefore, it is essential to consider the similarities that some of these problems have with SWCP, since some ideas and experiences can prove helpful in this research area.

Common elements between VRP and SWCP included having a homogenous fleet of vehicles, i.e., trucks. There are common elements between VRP and SWCP. For instance, VRP has a homogenous fleet of vehicles and a set of collection sites in a specific region. SWCP has a fleet of trucks and a set of containers in a city. A truck must collect a quantity within its capacity. Trucks should also start from a central depot, and after completing their collection route, they should end at the waste dumping area and then return to the central depot [10–12].

Truck assignment according to a time window was also considered in SWCP [13], i.e., the earliest and latest hours during which the collection service was permitted. Thus, SWCP could also be considered an extension of VRPTW. Researchers have also defined time-window constraints in connection with the containers' demands, such as in the work by [14–16].

To date, current research has focused on solution methods that tackle the objective function of VRPTW in the context of SWCP by minimising route length [17], waste content [18], or travel time [13, 15, 19]. However, this is only sometimes synonymous with minimising fuel use. Thus, previous studies would have been more valuable if they had focused on the negative impact on the environment of the waste collection process.

For example, the study by the authors of [20] presented a capacitive routing model for SWCP using a particle swarm optimisation (PSO) algorithm. A number of local improvement algorithms were also applied to improve the PSO's performance. The objective was to minimise operational costs while addressing environmental concerns. However, the study has only been conducted on VRP datasets rather than real-world scenarios.

The work by the authors of [21] minimised the total fuel consumption by using a tabu search (TS) with a random variable neighbourhood descent (RVND) procedure and an adaptive parallel route construction heuristic (APRCH). Using three different variations of the mathematical model, fuel consumption was reduced when time-window constraints were relaxed as a result of using fewer vehicles. A minimal-distance model performed better than a minimal-time model at reducing fuel usage. Nonetheless, a minimal-time model reduced the number of vehicles used, mainly when there were more nodes and tighter time windows. Hence, in this paper, the number of trucks used is considered in the objective function along with the waste load and the travelling time to reduce fuel consumption.

One potential area of solution methods for transportation management systems is heuristic algorithms. Efficient routing plays a critical role in optimising the delivery process and minimising costs. This was due to their potential to operate inside a constrained search area while still generating optimal solutions within a restricted computational timeframe [22]. Such as local search (LS) algorithms [12], adaptive large neighbourhood search (ALNS) heuristic, TS [23], simulated annealing (SA) approach [12], variable neighbourhood search (VNS), variable neighbourhood tabu search (VNTS) [15], and various initial solution strategies were also studied to improve the metaheuristic methods' performance such as the greedy heuristic [16], Clarke and Wright's savings algorithm, and the sweep algorithm [15].

For example, the work by [12] presented a priority-considered green VRP model for China's urban garbage collection and transportation problems. A local search hybrid algorithm was used to find the best solution using PSO as an initialisation strategy. Then, the SA was applied as a repair algorithm to the best initial solution. The method improved the quality of the results. Hence, local search algorithms were an efficient method to tackle waste collection.

In this paper, some elements of the VRPTW are considered while tackling the problem including capacity, demand, service times, and time window constraints. At the same time, we are aiming to minimise travel time, the number of trucks used, and fuel consumption. Local search methods have proven to be effective on VRP and VRPTW. This paper focuses on investigating the effect of minimising the objective function while reducing the computational times for SWCP using metaheuristic methods.

3. Problem Definition

To find feasible routes for the SWCP, this paper extends the mixed integer programming (MIP) model, described in detail in [4]. The objective function minimises the total travelling cost. Nevertheless, the work by [20, 21] has proposed methods for reducing fuel consumption including reducing the waste load for each route and the number of trucks used. In this paper, the objective function minimises the waste collection process's total effect on the environment along with the travel costs. In addition, constraints are adapted from the literature of VRP and VRPTW to suit the problem definition [13, 15, 19]. The following sections describe in detail the model's notations, as shown in Table 1.

SWCP is defined as a graph $G = (N, A)$, where $N = \{0,1,2\ldots,n+1\}$ is the set of nodes and $A = \{(i,j) \mid i,j \in N, i \neq j\}$ is the set of arcs. Each route is a sequence of nodes connected with arcs.

Node $0 \in N$ is the starting depot, i.e., camp, which serves as the base where the trucks start, while node $n+1 \in N$ is the dumping facility, i.e., dump, where all truck routes ultimately conclude. Other nodes are noted as a set $C = \{1,2,\ldots,n\}$, which are the locations of containers. Thus, for each arc $(i,j) \in A$, there are parameters d_{ij} and t_{ij} that denote the travel distance and time between nodes i to j, respectively.

In a route plan for SWCP, a set of trucks $K = \{1,2\ldots,k_{|K|}\}$ are assigned to collect waste from a set of containers $C = \{1,2,\ldots,n\}$, where each container has an amount of waste load q_i. The time duration required for a truck to fully empty the contents of a container i is defined as the service time, denoted as v_i. For each container, time windows are formally defined as intervals denoted by $[l_i, u_i]$, where l_i represents the minimum permissible time for visiting a container i may be visited. At the same time, u_i is the maximum permissible time for visiting a container i. Both capacity and time windows are considered hard constraints. Given the variable and parameter definitions, the SWCP model is formulated as follows:

Minimise,

$$z = c_d \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} T_{ij}x_{ijk} + \sum_{i\in N}\sum_{k\in K} c_t w_{ik} \sum_{i\in N}\sum_{k\in K} c_k y_{ik}, \quad (1)$$

$$\text{s.t.} \sum_{j\in N}\sum_{k\in K} x_{ijk} = 1, i \in C, \quad (2)$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0, h \in C, k \in K, \quad (3)$$

$$\sum_{i\in C} q_i \sum_{j\in N} x_{ijk} \leq Q y_{ik}, k \in K, \quad (4)$$

$$\sum_{j\in N\backslash\{0\}} x_{0jk} = 1, k \in K, \quad (5)$$

$$\sum_{i\in N\backslash\{n+1\}} x_{i,n+1,k} = 1, k \in K, \quad (6)$$

$$s_i + T_{ij} - U_0\left(1 - x_{ijk}\right) \leq s_j, i, j \in N, k \in K, \quad (7)$$

$$l_i \leq s_i \leq u_i, i \in N, \quad (8)$$

$$x_{ijk} \in \{0,1\}, i, j \in N, k \in K. \quad (9)$$

In the objective function shown in (1), there are three decision variables and four parameters. A binary variable x_{ijk} is equal to one if, while using a truck k, a container j is the successor of a container i; otherwise, zero. A continuous variable w_{ik} that indicates the amount of waste load k has when it arrives at i. A continuous variable y_{ik} indicates a truck k has arrived at i. For each container, i and container j, parameter T_{ij} is calculated as the total travel time t_{ij} added to the service time duration v_i and v_j. It is important to note that T_{ij} must take into account the service start time, denoted as s_i, and the time after leaving i and arriving to j, i.e., departure time, denoted as U_0. Where U_0 is a large number, defined as the sum of all u_i, i.e., the latest time window. All values are bounded by the time windows, as shown in (7) and (8). On the other hand, parameters c_d, c_t, and c_k are the costs of travelling per distance unit, per time unit, and per truck, respectively. These values act as weights to transform the objective function into a scalar objective function [24]. Hence, these weights tackle the model's tendency to have earlier visits, a lower waste load, and fewer trucks.

Constraints (2) and (3) guarantee that exactly one truck arrives/leaves one container. While constraint (4) ensures that the maximum capacity of a truck Q is never exceeded. Note that all vehicles have the same value of Q in this setting. Constraints (5) and (6) guarantee that each truck starts from the camp and finishes its route at the dump. Finally, the binary restriction on the route decision variables is shown in (9).

3.1. Case Study of SWCP. Instances of real-world Mecca SWCP are considered here. Mecca has visitors from around the world throughout the year to perform religious tasks and visit historic sites. Daily, a substantial quantity of waste is generated that requires effective management. During religious occasions, the typical amount of waste produced is estimated to be about three kilogrammes per individual. [25, 26]. At the same time, Mecca's resident population is increasing yearly, with 2,150 million residents reported in 2023 [27]. Furthermore, finding an optimal route for SWCP is critical. Mecca's urban areas with historical districts are challenging to manoeuvre, with narrow streets, and lots of pedestrians.

Table 1: Model notation table.

Notations	Description
Sets	
N	Set of nodes
A	Set of arcs
C	Set of containers
K	Set of waste collection trucks
Parameters	
d_{ij}	Integer parameter to indicate the travelling distance between node i and node j
t_{ij}	Integer parameter to indicate the travelling time between node i and node j
v_i	Integer parameter to indicate the service time for container i
T_{ij}	Integer parameter to indicate the travelling time between node i and node j in addition to v_i and v_j
c_d	Travel cost from node i to node j in regards to distance
c_t	Travel cost from node i to node j in regards to time
c_k	Travel cost from node i to node j in regards to the number of trucks
q_i	Container i maximum capacity
Q	Waste collection trucks maximum capacity
l_i	Earliest time to visit container i
u_i	Latest time to visit container i
Variables	
x_{ijk}	Decision variable to indicate that a node j is visited right after node i by a waste collection truck k
s_i	Integer variable to indicate the service starting time for node $i \in N$
w_{ik}	Integer variable to indicate the amount of load in a truck k when arriving at container i
y_{ik}	Integer variable to indicate the total number of trucks used when arriving at container i

The format of the problem instances is described in detail in [4]. Each container has a pair of coordinates, consisting of an x-coordinate representing the latitude location and a y-coordinate representing the longitude location. Therefore, the distance matrix is computed for every pair of containers. Since the matrix exhibits asymmetry, the distances between two containers are computed in kilometres. The distance matrix incorporates the earth's curvature to address real-world scenarios as it estimates the straight-line distance between geographic places. On the other hand, the lower and upper bounds of the service time parameters set a limit on service time. In the aforementioned scenarios, there are three distinct shifts that may be chosen: 4:00 a.m. to 12:00 p.m., 7:00 a.m. to 3:00 p.m., and 4:00 p.m. to 11:59 p.m. Furthermore, the current demand levels for nodes in these instances are currently unknown. Hence, the waste load was randomly generated within the range [1, 1.2] to simulate the stochastic nature of the demands.

The daily transportation routes for waste collection often include sequentially visiting each container throughout the city, regardless of whether they are filled with waste or not. In a deterministic scenario, it is assumed that the value of q_i remains constant for all containers. This means that all containers have the same capacity. Hence, the value of capacity Q remains constant across all instances. On average, the container capacity is around 20 tonnes, whereas the average truck capacity is approximately 60 tonnes.

4. Solution, Method, and Implementation

A metaheuristic method is a near-optimal algorithm designed to tackle complicated problems. Their wide use in recent years for combinatorial problems is due to their capability to address different parts of the problem and reach a feasible solution [28].

In this paper, the routing solver in the Google OR-tools solver is used as follows: (i) two initialisation heuristics generate good potential vehicle tours separately; (ii) three local search metaheuristics methods improve the generated solutions separately to guide the search; and (iii) a constraint programming (CP) engine improves the quality of the best solution. The aim is to compare their performance in the optimisation process for the SWCP model. The following are the details of the implemented methods.

4.1. Initial Solution. To obtain approximations of solutions to combinatorial optimisation problems, local search algorithms begin with a candidate solution and then iteratively navigate to a neighbouring solution. The collection of all potential solutions close to the current solution is known as a *neighbourhood* [15].

The move to the following solution is defined by the *acceptance movement*. The action that reduces the objective value the most is selected at each stage of the search. The method cannot reduce the number of routes if no feasible solution is found.

Generally, a construction method is used to produce an initial feasible solution, and then a heuristics is used to maintain the feasibility. Local search algorithms are used to escape the local minima as a result of guiding the search [15].

Two heuristic methods are applied to find the first solution for the SWCP: Clarke and Wright's algorithm as well as the nearest neighbour algorithm.

The Clarke and Wright's algorithm is the most widely known heuristic for the VRP. It was first proposed by [29]. It begins with each truck serving two containers in a single trip. Then, the savings cost is calculated $s(i, j) = d(D, i) + d(D, j) - d(i, j)$ for every pair of containers (i, j) by satisfying the demands. Then, the savings $s(i, j)$ are ranked in descending order in the savings list. For the savings $s(i, j)$ under consideration, the link is included if there are no route violations of the capacity restrictions and there exists one of the following cases:

(i) If neither i nor j has already been assigned to a route, a new route with both i and j is started.

(ii) Or, if either i or j has already been included in an existing route and that container is not interior to that route, the link between i and j is added to that same route.

(iii) Or, both i and j have already been included in two different existing routes, and neither container is interior to its route, in which case the two routes are merged. The algorithm merges the containers in the savings list corresponding to the highest saving value without exceeding the capacity restriction.

The algorithm continues to execute until no more route additions or merges are available. Note that each node has at least one neighbouring node. Thus, the arc cost used to calculate the saving value has a coefficient of 1 [30].

The nearest neighbour algorithm generates routes one at a time, by attaching the container j, which is the container i's closest unrouted neighbour in terms of the objective value, to the container i.

It starts at the starting camp in the first phase and connects to an unrouted node after that. The chosen node must provide the cheapest route arc to initialise a single route. The procedure connects the current node to the following unrouted node in step two until the last node is included in the route. A new route is initialised if all nodes have yet to be incorporated into the routes. [30].

The neighbourhood is assessed following the generation of the current solution's neighbourhood. The local search employs a move strategy to choose at most one solution from the neighbourhood as the new current solution [11]. All techniques switch between solutions using the group of operators stated next [30].

4.2. Operators. During the search, the algorithm uses the following operators:

(i) *Operators for the allocation of customers to routes (inter-route optimisation)*

  (a) *2-opt* replaces two edges from the route under consideration with two new edges to create a new route [31].

  (b) *Or opt* improves the existing route by first changing the placement of a sequence of three successive vertices until no more improvement is possible. Next, the same procedure is repeated on a chained series of two successive vertices, and finally, on single vertices [32].

  (c) *Lin Kernighan (LK) operator* searches for a shorter route in a neighbourhood, then uses that shorter route as a starting point and repeats the procedure until it reaches a local minimum. Since LK is an adaptive approach, there is no limit on how many edges can be replaced at one time [32].

(ii) *Operators for the optimisation of each route (intra-route optimisation)*

  (a) *Relocate operator*, where a single container is removed from its route and placed on another route [30].

  (b) *Swap-exchange operator* exchanges two containers from two different routes [30].

  (c) *Cross-exchange operator* exchanges substrings connected to the camp. As a result, the substrings may have to be inverted [33].

5. Local Search-Based Metaheuristic Methods

Combinatorial problems are solved using well-known metaheuristics, rather than by customising the approach for each problem. Hence, this paper implemented three well-known local search metaheuristic methods, presented by [28], to tackle SWCP.

5.1. Guided Local Search. The guided local search (GLS) is a metaheuristic algorithm [34]. GLS escapes local minima by penalising a solution with the distance arc that it considers should not occur in a near-optimal solution, based on the experience the search has gained [35]. Solution features are defined if there is a direct arc between two containers. Features are used to distinguish between solutions with various properties. This allows for identifying and avoiding regions of similarity around local optimums. Algorithm 1 gives an outline of the method [28].

In this study, an edge is penalised with a $\lambda = 0.1$ coefficient during the search, and the interroute operators are applied to this edge. Thus, only moves that begin with the removal of the relevant edge are taken into account. The move that decreases the objective value by the most amount is used as the move [11].

Then, the intraroute operator is used to optimise the changed routes. The combination of inter- and intraroute optimisation is applied in each iteration until the solution is not improved or the time limit is exceeded [30].

5.2. Tabu Search. A local search metaheuristic known as Tabu Search (TS) was first published by Glover [36]. In order to explore the search space, TS iteratively moves from one solution to the next until it locates the best solution in a subset of its neighbourhood. The tabu list keeps track of recent actions or visits to solutions. The algorithm is forbidden from making any of the moves on the tabu list. This forces the algorithm to look at alternative options. Tabu factor is an acceleration parameter to explore different moves in the neighbourhood. The method presented by [28] is outlined in Algorithm 2.

The interroute operators are applied to the current solution after each iteration. The best solution is selected. Either a tabu or nontabu solution exists. It is tabu if the action that results in the solution is also tabu. However, an aspiration criterion may override the tabu status (the best total cost seen so far). When the tabu status expires or is overridden by the aspiration criterion [30], the opposite move is then classified as tabu for the next $p = 5$ iterations.

5.3. Simulated Annealing. Simulated annealing (SA) was introduced by [37]. Material is heated to a high temperature in a technique modelled after the physical annealing process. The temperature (T) is gradually lowered to get the material to a state of minimal energy, i.e., reach a thermodynamic equilibrium, avoiding local energy minima in the process. Similar to the system's energy, the objective function in the optimisation context must be minimised. SA is a metaheuristic that attempts to escape local minima by exploring the solution space. Improving candidate solutions is always accepted, while nonimproving candidate solutions are accepted with a certain probability. In this case, the control parameter (T) lowers gradually and tends to be zero according to a deterministic cooling schedule. The search is initially diverse, and the solution space is explored more globally. As the value of T decreases, the search becomes more focused on specific areas of the search space. The approach presented by [28] is described in the Algorithm 3.

6. Computational Results

Using default settings, the implementation has been evaluated using Windows 10 and 64 bit operating system with Gurobi Solver 9.5 [38] and OR-Tools solver for vehicle routing problems [39]. The model is coded in C# in the MS Visual Studio environment. To assess the results, each run was executed 8 times, seeded with the same random number, and all methods had the same amount of computation time.

OR-Tools are an open-source solver developed by Google. All heuristic methods generate a solution for the Gurobi solver, which enables the solver to escape a local minimum, i.e., a solution shorter than all nearby routes but not the global minimum. After moving away from the local minimum, the solver continues the search, thus, allowing iterative improvements to the solution.

The parameters used in this paper were as follows: The tabu factor for the TS was 0.8, the keep iterations were 10, and the forbid iterations were 10. $T = 100$ for the SA, where T at iteration i is $T_i = T_0/i$.

Fixed parameters were q_i and Q, i.e., container and truck capacity, respectively. When a truck is empty, it weighs 20 tonnes. Random variables, demand, were seeded to be able to range between 0 and 10 tonnes. In order to conduct the experiments with the random variables, eight trials were executed, and the average computational times of these trials were used.

Table 2 presents the results of the IP solver on 13 different instances. For each problem instance, the table shows the total distance in *Km*, noted as *Distance*; the total travelling time in *Seconds*, noted as *Time*; the cumulative vehicle load in *Tonnes*, noted as *Veh. Load*; the number of trucks used per route, noted as *Routes*, solution quality, noted as *Obj*; and the computational time in *Seconds*, noted as *Cpt*, in which the best solution was found, and the gap to the optimal, noted as *Gap%*.

There were no violations. Hence, the proposed model obtained optimal solutions for all instances within an acceptable amount of time. Additionally, minimising the number of trucks used means less fuel consumption and, henceforth, fewer daily routes. Thus, not all trucks were used. On the other hand, the size of the problem has affected the objective values. The larger the problem, the more time was required to find an optimal solution. As in the cases of 15–50, 20–60, 30–60, 30–70, and 30v75, the model obtained optimal solutions that required more computational time. The more containers added, the greater the computational time. This occurrence was due to the fact that succeeding nodes have higher load levels. Hence, the model prioritised the collection of lighter nodes first and heavier nodes afterwards to minimise the overall weight. Therefore, a number of nodes were disregarded during the execution of the algorithm and thereafter needed to be added at a significant cost. Therefore, the model was examined for further experiments using metaheuristic methods.

Table 3 presents the results of the three local search methods (GLS, TS, and SA) with two different initialisation algorithms (Clarke and Wright's and the nearest neighbour) using the MIP model. For each problem instance, the table shows the total distance in *Km*, noted as *Distance*; the total travelling time in *Seconds*, noted as *Time*; the cumulative vehicle load in *Tonnes*, noted as *Veh. Load*; the number of trucks used per route, noted as *Routes*, solution quality, noted as *Obj*; the average computational time in *Seconds* for all runs, noted as *Cpt*, in which the best solution was found; and the gap to the optimal, noted as *Gap%*. The best computational times are highlighted in *bold*.

All methods have obtained optimal solutions. Local search methods obtained the results faster when using the nearest neighbour initialisation algorithm on 77% of all the problems.

Each subfigure in Figure 1 corresponds to a problem instance to illustrate the overall computational time (in seconds) used. Search algorithms using the Clarke and Wright's strategy are indicated with*, while algorithms using the nearest neighbour algorithm are indicated by ('). Black solid bars when using GLS*, grey bars with stripped lines when using GLS', blue bars with dots when using TS*, red bars with right inclined lines when using TS', green bars with

Initialise penalties $p_i := 0$ and costs c_i
Generate initial solution x
while stopping criterion is not met **do**
  Find a local minimum x^* of the augmented objective function using local search starting from x
  Compute the utility u_i of each feature F_i of x
  $j \leftarrow \text{argmax}_i\ u_i$
  $p_j \leftarrow p_i + 1$
  $x \leftarrow x^*$
end while
return best solution found

Algorithm 1: Guided local search.

Generate initial solution x
Initialise empty tabu list
while stopping criterion is not met **do**
  Generate new solution $x' \in N(x)$ not in the tabu list
  $x \leftarrow x'$
  Update tabu list
end while
return best solution found

Algorithm 2: Tabu search.

Generate initial solution x
Set initial temperature T
while stopping criterion is not met **do**
  repeat
    Generate new solution $x' \in N(x)$
    if $f(x') \leq f(x)$ **then**
      $x \leftarrow x'$
    else
      $x \leftarrow x'$ with probability $\exp - f(x') - f(x)/T$
    end if
  until thermodynamic equilibrium
  Decrease T
end while
return best solution found

Algorithm 3: Simulated annealing.

Table 2: Objective values for the optimal solution obtained by the MIP model.

Ins.		Solutions obtained by solver						
#Veh	#Cont.	Distance	Time	Veh. load	Routes	Obj	Cpt	% gap
3	5	125.58	73.58	16.10	1.00	77.83	0.10	0.0%
5	10	196.98	82.15	29.51	1.00	87.95	0.13	0.0%
5	15	199.61	85.31	43.06	1.00	92.11	0.25	0.0%
6	10	229.11	82.15	29.51	1.00	87.95	0.14	0.0%
9	15	328.13	85.31	43.06	1.00	92.11	0.24	0.0%
9	20	330.82	88.54	50.77	1.00	95.57	0.29	0.0%
10	25	386.49	116.79	66.51	1.00	124.25	0.55	0.0%
10	35	385.69	154.38	85.08	2.00	162.37	1.23	0.0%
15	50	566.07	178.05	117.55	2.00	186.53	2.43	0.0%
20	60	740.06	194.07	139.19	2.00	202.68	34.27	0.0%
30	60	1061.36	194.07	139.19	2.00	202.67	21.88	0.0%
30	70	1073.77	208.96	163.67	2.00	217.66	1891.78	0.0%
35	75	1235.82	210.64	173.56	2.00	219.50	5888.39	0.0%

Table 3: Comparison of solution objective values for the optimal solution, obtained by GLS, TS, and SA, using the Clarke and Wright's algorithm and the nearest neighbour algorithm.

	Ins.		Clarke and Wright's algorithm							Nearest neighbour algorithm						
	#Veh	#Cont.	Distance	Time	Veh. load	Routes	Obj	Cpt	% gap (%)	Distance	Time	Veh. load	Routes	Obj	Cpt	% gap (%)
	3	5	125.58	73.58	16.10	1.00	77.83	**0.19**	0.0	125.58	73.58	16.10	1.00	77.83	0.26	0.0
	5	10	196.98	82.15	29.51	1.00	87.95	**0.24**	0.0	196.98	82.15	29.51	1.00	87.95	0.28	0.0
	5	15	199.61	85.31	43.06	1.00	92.11	**0.28**	0.0	199.61	85.31	43.06	1.00	92.11	0.44	0.0
	6	10	229.11	82.15	29.51	1.00	87.95	0.42	0.0	229.11	82.15	29.51	1.00	87.95	0.30	0.0
	9	15	328.13	85.31	43.06	1.00	92.11	0.63	0.0	328.13	85.31	43.06	1.00	92.11	**0.34**	0.0
	9	20	330.82	88.54	50.77	1.00	95.57	1.03	0.0	330.82	88.54	50.77	1.00	95.57	0.42	0.0
GLS	10	25	386.49	116.79	66.51	1.00	124.25	0.93	0.0	386.49	116.79	66.51	1.00	124.25	0.57	0.0
	10	35	385.69	154.38	85.08	2.00	162.37	1.72	0.0	385.69	154.38	85.08	2.00	162.37	**0.91**	0.0
	15	50	566.07	178.05	117.55	2.00	186.53	2.04	0.0	566.07	178.05	117.55	2.00	186.53	**1.81**	0.0
	20	60	740.06	194.07	139.19	2.00	202.68	42.52	0.0	740.06	194.07	139.19	2.00	202.68	**32.03**	0.0
	30	60	1061.36	194.07	139.19	2.00	202.67	21.41	0.0	1061.36	194.07	139.19	2.00	202.67	**21.00**	0.0
	30	70	1073.77	208.96	163.67	2.00	217.66	1751.72	0.0	1073.77	208.96	163.67	2.00	217.66	1807.45	0.0
	35	75	1235.82	210.64	173.56	2.00	219.50	6528.07	0.0	1235.82	210.64	173.56	2.00	219.50	**3962.95**	0.0
	3	5	125.58	73.58	16.10	1.00	77.83	0.36	0.0	125.58	73.58	16.10	1.00	77.83	0.28	0.0
	5	10	196.98	82.15	29.51	1.00	87.95	0.36	0.0	196.98	82.15	29.51	1.00	87.95	0.27	0.0
	5	15	199.61	85.31	43.06	1.00	92.11	0.54	0.0	199.61	85.31	43.06	1.00	92.11	0.29	0.0
	6	10	229.11	82.15	29.51	1.00	87.95	0.53	0.0	229.11	82.15	29.51	1.00	87.95	0.28	0.0
	9	15	328.13	85.31	43.06	1.00	92.11	0.60	0.0	328.13	85.31	43.06	1.00	92.11	0.38	0.0
	9	20	330.82	88.54	50.77	1.00	95.57	0.66	0.0	330.82	88.54	50.77	1.00	95.57	0.41	0.0
TS	10	25	386.49	116.79	66.51	1.00	124.25	1.03	0.0	386.49	116.79	66.51	1.00	124.25	0.58	0.0
	10	35	385.69	154.38	85.08	2.00	162.37	1.46	0.0	385.69	154.38	85.08	2.00	162.37	0.96	0.0
	15	50	566.07	178.05	117.55	2.00	186.53	1.89	0.0	566.07	178.05	117.55	2.00	186.53	1.89	0.0
	20	60	740.06	194.07	139.19	2.00	202.68	41.06	0.0	740.06	194.07	139.19	2.00	202.68	35.63	0.0
	30	60	1061.36	194.07	139.19	2.00	202.67	21.24	0.0	1061.36	194.07	139.19	2.00	202.67	21.15	0.0
	30	70	1073.77	208.96	163.67	2.00	217.66	1430.72	0.0	1073.77	208.96	163.67	2.00	217.66	1573.23	0.0
	35	75	1235.82	210.64	173.56	2.00	219.50	4011.22	0.0	1235.82	210.64	173.56	2.00	219.50	4220.71	0.0
	3	5	125.58	73.58	16.10	1.00	77.83	0.37	0.0	125.58	73.58	16.10	1.00	77.83	0.24	0.0
	5	10	196.98	82.15	29.51	1.00	87.95	0.34	0.0	196.98	82.15	29.51	1.00	87.95	0.24	0.0
	5	15	199.61	85.31	43.06	1.00	92.11	0.41	0.0	199.61	85.31	43.06	1.00	92.11	0.30	0.0
	6	10	229.11	82.15	29.51	1.00	87.95	0.33	0.0	229.11	82.15	29.51	1.00	87.95	**0.25**	0.0
	9	15	328.13	85.31	43.06	1.00	92.11	0.34	0.0	328.13	85.31	43.06	1.00	92.11	0.35	0.0
	9	20	330.82	88.54	50.77	1.00	95.57	0.37	0.0	330.82	88.54	50.77	1.00	95.57	**0.36**	0.0
SA	10	25	386.49	116.79	66.51	1.00	124.25	0.63	0.0	386.49	116.79	66.51	1.00	124.25	**0.54**	0.0
	10	35	385.69	154.38	85.08	2.00	162.37	0.93	0.0	385.69	154.38	85.08	2.00	162.37	0.99	0.0
	15	50	566.07	178.05	117.55	2.00	186.53	1.87	0.0	566.07	178.05	117.55	2.00	186.53	1.82	0.0
	20	60	740.06	194.07	139.19	2.00	202.68	34.11	0.0	740.06	194.07	139.19	2.00	202.68	35.59	0.0
	30	60	1061.36	194.07	139.19	2.00	202.67	21.48	0.0	1061.36	194.07	139.19	2.00	202.67	21.06	0.0
	30	70	1073.77	208.96	163.67	2.00	217.66	1386.93	0.0	1073.77	208.96	163.67	2.00	217.66	**1226.45**	0.0
	35	75	1235.82	210.64	173.56	2.00	219.50	4759.30	0.0	1235.82	210.64	173.56	2.00	219.50	7195.00	0.0

The best computational times are highlighted in bold.

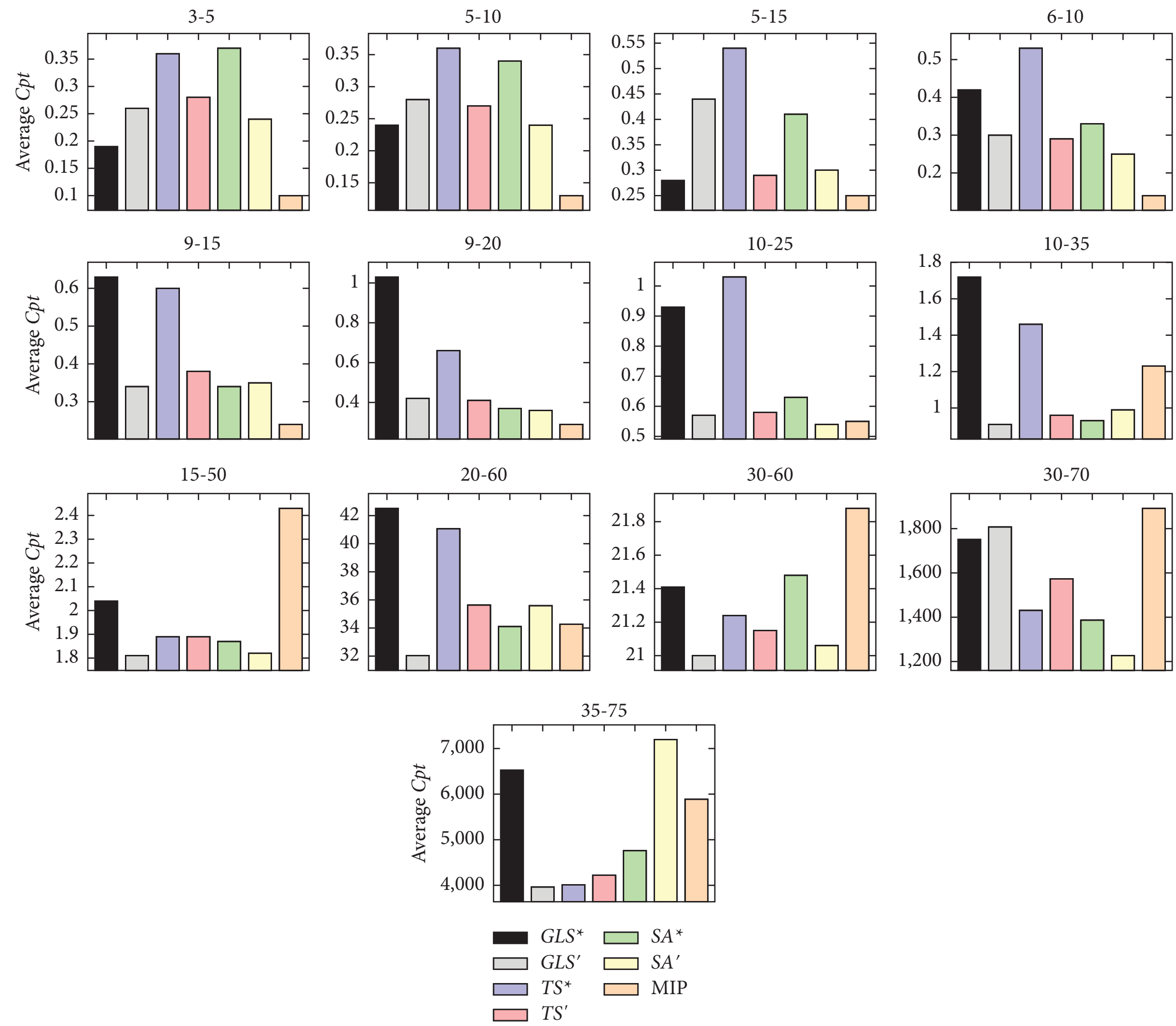

Figure 1: Computational times (in seconds).

left inclined lines when using SA*, yellow bars with a grid when using SA, and finally orange bars with vertical lines when using MIP. The lower the bar, the better, i.e., the less computational time.

Computational times were reduced compared to the solver on instances 10–25, 10–35, 15–50, 20–60, 30–60, 30–70, and 35–75. This was due to the algorithm's greedy nature, which allowed fast computation of the operational cost. On the other hand, results obtained by the Clarke and Wright's algorithm as an initialisation strategy were only faster than the other algorithms on instances 3–5, 5–10, and 5-15. Still, the computation times obtained by the solver on those instances were better. Hence, finding a feasible initial solution, especially for large problems, to an SWCP is critical.

Different algorithms have different strategies for route construction. Nevertheless, a better initial solution will help the algorithm search for a better solution with faster computational times. This was clear with both initialisation strategies: SA finds the optimal solutions faster than TS, especially when combined with the nearest neighbour algorithm. This was due to the number of neighbourhoods considered for each move. In SA, only one neighbourhood per move. However, in TS, n number of neighbourhoods were considered. Thus, more time was spent exploring the neighbouring solutions.

GLS with the Clarke and Wright algorithm was the slowest out of the six methods, with 38% of all problem instances. In contrast, GLS obtained the highest number of best computation times, with 69% overall solutions using the two initialisation methods. Combined with the Clarke and Wright algorithm, GLS was faster for small problems. Combined with the nearest neighbour algorithm, GLS was faster for larger problems. This was due to the GLS penalisation process. Each time the GLS was trapped in a local minimum, the penalty parameter increased. Hence, unfavoured features were penalised, and solutions with these features were avoided, which improved the efficiency and robustness of the underlying local search algorithms. Hence, GLS was a very competitive algorithm for this problem.

7. Conclusion

This paper investigated the performance of well-known metaheuristic methods to generate feasible solutions to SWCP while considering all conditions. Two initialisation strategies were used: Clarke and Wright's and the nearest neighbour algorithms. After the first route was generated, three well-known metaheuristic methods were used for neighbourhood exploration, including guided local search (GLS), tabu search (TS), and simulated annealing (SA). The aim was to minimise the objective function by reducing travel times, waste loads, and the number of trucks used while reducing computational time.

These metaheuristic methods significantly improved the efficiency of the route optimisation process, leading to reduced fuel consumption and better overall performance. Optimal results were found faster than using only an IP solver, especially for large instances. In this paper, using local search methods with the nearest neighbour algorithm was more efficient since it outperformed the Clarke and Wright method in most instances. Overall, GLS outperformed TS and SA most of the time. The findings here can be applied to improve operations in Saudi Arabia's waste management sector.

Using Google OR-tools allowed modelling the problem while using different search strategies. However, constraint handling methods were not incorporated into the existing OR-tools search strategies, and only general algorithms were used. This was enough for the instances in this paper; however, problem-specific information must be used to solve problems with more than 100 vehicles. In order to further optimise the search process, a hybrid algorithm can be applied to enhance the local search and population management strategy, such as hybridising metaheuristics methods with genetic algorithms (GAs) [40] or ant colony optimisation (ACO) algorithm [41]. The application of problem-specific knowledge in the design of heuristics can be beneficial in solving larger problem instances [42]. Also, hybridisation methods that consider problem-specific knowledge are required to improve solution quality [12, 43]. Thus, another future direction can be utilising a hybrid heuristic algorithm to reduce the number of nodes and vehicles by splitting the problem into manageable clusters and solving the routing problem within each cluster. This method combines the intensification process by using LS operators with the diversification of using constructive heuristics [44]. The algorithm adopted in this study can be compared with the hybrid algorithms to evaluate their effectiveness.

Data Availability

The data used to support the findings of this study are available upon request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The researchers would like to acknowledge the Deanship of Scientific Research, Taif University for funding this work.

References

[1] M. Hannan, R. A. Begum, A. Q. Al-Shetwi et al., "Waste collection route optimisation model for linking cost saving and emission reduction to achieve sustainable development goals," *Sustainable Cities and Society*, vol. 62, Article ID 102393, 2020.

[2] B. Maina, "The challenges of solid waste management in sustainable urban planning," in *Proceedings of the 11th International Conference on Urban Regeneration and Sustainability*, Georgia Institute of Technology, Alicante, Spain, July 2016.

[3] Epa, "Fast facts on transportation greenhouse gas emissions," 2023, https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions.

[4] H. Algethami and G. T. Alhothali, "Waste collection optimisation: a path to a green and sustainable city of makkah," *Logistics*, vol. 7, no. 3, p. 54, 2023.

[5] J. Lella, V. R. Mandla, and X. Zhu, "Solid waste collection/transport optimization and vegetation land cover estimation using geographic information system (gis): a case study of a proposed smart-city," *Sustainable Cities and Society*, vol. 35, pp. 336–349, 2017.

[6] N. Radwan, N. A. Khan, and R. A. G. Elmanfaloty, "Optimization of solid waste collection using rsm approach, and strategies delivering sustainable development goals (sdg's) in jeddah, Saudi Arabia," *Scientific Reports*, vol. 11, no. 1, pp. 16612–12, 2021.

[7] J. Hartmanis, "Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson)," *SIAM Review*, vol. 24, no. 1, p. 90, 1982.

[8] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.

[9] M. Hannan, M. Hossain Lipu, M. Akhtar et al., "Solid waste collection optimization objectives, constraints, modeling approaches, and their challenges toward achieving sustainable development goals," *Journal of Cleaner Production*, vol. 277, Article ID 123557, 2020.

[10] O. Braysy and M. Gendreau, "Vehicle routing problem with time windows, part i:route construction and local search algorithms," *Transportation Science*, vol. 39, no. 1, pp. 104–118, 2005.

[11] F. Arnold and K. Sörensen, "Knowledge-guided local search for the vehicle routing problem," *Computers and Operations Research*, vol. 105, pp. 32–46, 2019.

[12] H. Wu, F. Tao, and B. O. Yang, "Optimization of vehicle routing for waste collection and transportation," *International Journal of Environmental Research and Public Health*, vol. 17, no. 14, Article ID 4963, 2020.

[13] B. I. Kim, S. Kim, and S. Sahoo, "Waste collection vehicle routing problem with time windows," *Computers and Operations Research*, vol. 33, no. 12, pp. 3624–3642, 2006.

[14] B. M. Ombuki-Berman, A. Runka, and F. T. Hanshar, "Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms," in *Proceedings of the Third IASTED International Conference on Computational Intelligence*, pp. 91–97, Citeseer, Banff, Canada, May 2007.

[15] A. M. Benjamin, *Metaheuristics for the waste collection vehicle routing problem with time windows*, Ph.D. thesis, Brunel

University, School of Information Systems, Computing and Mathematics, Uxbridge, UK, 2011.

[16] K. Buhrkal, A. Larsen, and S. Ropke, "The waste collection vehicle routing problem with time windows in a city logistics context," *Procedia-Social and Behavioral Sciences*, vol. 39, pp. 241–254, 2012.

[17] R. Islam and M. S. Rahman, "An ant colony optimization algorithm for waste collection vehicle routing with time windows, driver rest period and multiple disposal facilities," in *Proceedings of the 2012 International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 774–779, IEEE, Dhaka, Bangladesh, May 2012.

[18] M. Dotoli and N. Epicoco, "A vehicle routing technique for hazardous waste collection," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9694–9699, 2017.

[19] B. Agacsapan and S. N. Cabuk, "Determination of suitable waste transfer station areas for sustainable territories: eskisehir case," *Sustainable Cities and Society*, vol. 52, Article ID 101829, 2020.

[20] M. 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, 2018.

[21] L. Jin, "Vehicle routing problem with time windows for reducing fuel consumption," *Journal of Computers*, vol. 7, no. 12, pp. 3020–3027, 2012.

[22] S. P. Gayialis and I. P. Tatsiopoulos, "Design of an it-driven decision support system for vehicle routing and scheduling," *European Journal of Operational Research*, vol. 152, no. 2, pp. 382–398, 2004.

[23] J. F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.

[24] J. De Armas, H. Ramalhinho, E. Lalla-Ruiz, B. Melian-Batista, and M. Moreno-Vega, "Operations research for transportation and sustainable development," *Journal of Advanced Transportation*, vol. 2019, Article ID 4146362, 2 pages, 2019.

[25] I. Elgammal and G. T. Alhothali, "Towards green pilgrimage: a framework for action in makkah, Saudi Arabia," *The international journal of religious tourism and pilgrimage*, vol. 9, no. 1, pp. 39–57, 2021.

[26] Azalghamdi, "Population in makkah region by gender, age group, and nationality (saudi/non-saudi)," 2018, https://www.stats.gov.sa/en/5723.

[27] Macrotrends, "Population-level and growth rate for the mecca, Saudi Arabia metro area from 1950 to 2023," 2023, https://www.macrotrends.net/cities/22426/mecca/population.

[28] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.

[29] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.

[30] P. Määttä, *Comparison of solution methods for the dynamic pickup and delivery problem with time windows*, Aalto University, Ph.D. thesis, 2021.

[31] J. J. Bentley, "Fast algorithms for geometric traveling salesman problems," *ORSA Journal on Computing*, vol. 4, no. 4, pp. 387–411, 1992.

[32] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[33] P. H. V. Penna, A. Subramanian, and L. S. Ochi, "An iterated local search heuristic for the heterogeneous fleet vehicle routing problem," *Journal of Heuristics*, vol. 19, no. 2, pp. 201–232, 2013.

[34] C. Voudouris, E. Tsang, and A. Alsheddy, *Guided Local Search*, Springer, Berlin, Germany, 2010.

[35] P. Kilby, Patrick Prosser, and P. Shaw, "Guided local search for the vehicle routing problem with time windows," in *Metaheuristics*, pp. 473–486, Springer, Berlin, Germany, 1999.

[36] F. Glover, "Tabu search: a tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

[37] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[38] Gurobi, "Gurobi 7.5 user's manual," 2017, https://www.gurobi.com/documentation/current/refman/index.html.

[39] Google, "Or-tools; google developers," 2023, https://developers.google.com/optimization/.

[40] B. Rezaei, F. G. Guimaraes, R. Enayatifar, and P. C. Haddow, "Combining genetic local search into a multi-population imperialist competitive algorithm for the capacitated vehicle routing problem," *Applied Soft Computing*, vol. 142, Article ID 110309, 2023.

[41] S. Akpinar, "Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 61, pp. 28–38, 2016.

[42] F. Arnold and K. Sörensen, "What makes a vrp solution good? the generation of problem-specific knowledge for heuristics," *Computers and Operations Research*, vol. 106, pp. 280–288, 2019.

[43] M. A. Dulebenets, "An adaptive polyploid memetic algorithm for scheduling trucks at a cross-docking terminal," *Information Sciences*, vol. 565, pp. 390–421, 2021.

[44] M. Mestria, "A hybrid heuristic algorithm for the clustered traveling salesman problem," *Pesquisa Operacional*, vol. 36, no. 1, pp. 113–132, 2016.