Computers & Industrial Engineering 161 (2021) 107618

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Optimizing a waste collection system with solid waste transfer stations

Gianpaolo Ghiani, Andrea Manni, Emanuele Manni *, Valentino Moretto

Department of Engineering for Innovation, University of Salento, Lecce, LE 73100, Italy

ARTICLE INFO

Keywords:
Municipal solid waste management
Transfer stations
Vehicle routing with transshipment
Heuristics
Matheuristics

ABSTRACT

In recent years, stricter environmental regulations, as well as an increased public concern, have progressively forced new landfills to be located more and more away from urban centers. This has stimulated the use of solid waste transfer stations, where the solid waste is transferred from small collection vehicles to large transportation vehicles. In this paper, we tackle the problem of determining the routes for both collection and transportation vehicles, as well as their synchronization at the transfer stations. We divide the problem into two phases and propose an exact approach utilizing a mathematical formulation, as well as a constructive heuristic and a matheuristic for the first phase, and a heuristic approach for the second phase. Computational results show that the approach combining the matheuristic for the collection phase with the heuristic for the transportation phase is able to achieve consistent reductions in terms of number of collection vehicles needed.

1. Introduction

In recent years, Europe has made substantial progress towards a more sustainable management of Municipal Solid Waste (MSW). The average amount of waste per capita has been reduced from 527 kg in 2002 to 492 kg per year in 2018 even though this amount varies significantly across EU Member States (Eurostat statistics explained, 2018). MSW management requires an extremely complex structure that implies the execution of many activities, concerning several stages of the waste life-cycle: collection, (possibly) transformation, and disposal. In turn, each activity involves taking a number of decisions at the strategic, tactical, and operational levels. Examples of such decisions are: selecting waste treatment sites and/or landfills (Wang, Qin, Li, & Chen, 2009), locating collection sites (Ghiani, Laganà, Manni, & Triki, 2012), zoning the service territory into districts (Ghiani, Manni, Manni, & Toraldo, 2014; Mostafayi Darmian, Moazzeni, & Hvattum, 2020), selecting collection days for each zone and for each waste type (Chu, Labadi, & Prins, 2006), determining fleet composition and collection vehicles' routing and scheduling (Ghiani, Guerriero, Manni, Manni, & Potenza, 2013). The reader is referred to Ghiani, Laganà, Manni, Musmanno, and Vigo (2014) for a survey of the most relevant issues.

In this paper, we consider the collection stage, and focus on the case with intermediate solid waste transfer stations (SWTS). Indeed, during the last few decades, several factors, such as more rigorous environmental regulations, together with an increased public concern, have progressively forced new landfills to be located more and more away from urban centers. This increased distance needed to reach the landfills, together with the fact that often the access to city centers is permitted only to specific types of vehicles (such as electric vehicles) with a limited capacity and (battery) autonomy, has given an incentive to the usage of SWTS, where solid waste is transferred from small collection vehicles to larger, long-haul transportation vehicles. The presence of SWTS brings to an overall problem that requires to determine: (i) the route of each small collection vehicle, including the arrival time at the SWTS for each leg of the route; (ii) the routes of the large vehicles (from the landfill to the SWTS and vice versa) for which the visits at the SWTS must be synchronized with those of the small vehicles. For phase (i) we propose both an exact and a heuristic approach, utilizing a mathematical formulation and a constructive procedure, respectively. Moreover, we also illustrate a procedure made up of a combination of both approaches, resulting in a matheuristic. Finally, to solve phase (ii) we present a heuristic approach.

The problem of determining the routes for both types of vehicles can be cast as a Vehicle Routing Problem with Transshipment (VRPT), in which the presence of transshipment locations allows vehicles to stop to adjust and/or transfer their transport loads. In addition, drivers are allowed to switch their vehicles or fresh/rested drivers may replace the tired drivers. The increased flexibility given by the transshipment locations raises additional modeling challenges for the traditional Vehicle Routing Problem (in which the problem is that of "simply" determining the vehicle routes), mainly because the spatial and time operations involving the vehicles must be properly matched and synchronized.

* Corresponding author.
E-mail addresses: gianpaolo.ghiani@unisalento.it (G. Ghiani), andrea.manni@unisalento.it (A. Manni), emanuele.manni@unisalento.it (E. Manni).

https://doi.org/10.1016/j.cie.2021.107618
Received 4 February 2021; Received in revised form 6 July 2021; Accepted 11 August 2021
Available online 14 August 2021
0360-8352/

Such aspects are thoroughly examined and discussed in the review by Drexl (2012).

In the context of using transshipment locations for waste collection, a seminal paper is that of De Rosa, Improta, Ghiani, and Musmanno (2002), in which the authors introduce the Arc Routing and Scheduling Problem with Transshipment and propose a lower bound, based on a relaxation of an integer linear formulation of the problem, as well as a tailored tabu search heuristic. Later, Ghiani, Guerriero, Laporte, and Musmanno (2004) develop three heuristics (a constructive procedure based on a partitioning approach, and two tailored tabu search procedures) for a similar problem, namely the Arc Routing Problem with Intermediate Facilities under Capacity and Length Restrictions. For the same problem, Willemse and Joubert (2016) devise four constructive heuristics to compute feasible solutions with a main goal to either minimize the total cost or to minimize the fleet size. Another work is due to Komilis (2008), who propose two mixed-integer linear programming models to optimize the haul and transfer of municipal solid waste prior to landfilling. The models are illustrated on a case study related to the municipality of Athens. More recently, Das and Bhattacharyya (2015) present both a mixed integer program and a heuristic solution for the waste collection and transportation problem, whereas for a similar problem Rízvanoğlu, Kaya, Ulukavak, and Yeşilnacar (2019) develop a linear programming model, tested on the Şanliurfa Province, Turkey.

In addition, there are other works about the usage of SWTS, although they mainly focus on the analysis of their economical impact. Examples of such papers are De Oliveira, De Oliveira, and Rueda (2017) and Höke and Yalcinkaya (2021).

The contribution of our paper is the determination of the routes for both sets of collection and transportation vehicles, as well as their synchronization at the transfer stations. As briefly mentioned before, these goals are achieved through the development of a mathematical model and two heuristic approaches for the collection phase, and a heuristic approach for the transportation phase.

The remainder of the paper is organized as follows. In Section 2 we describe in details the problem we are studying, whereas our approaches for the collection and the transportation phases are presented in Sections 3 and 4, respectively. Finally, in Section 5 we illustrate the results of the computational experiments we have performed to assess our approaches, and conclusions follow in Section 6.

2. Problem description

The problem we face in this paper is characterized by a number of collection points (or collection zones), where each of them represents a cluster of citizens, grouped according to their position. A cluster may include a block, or all the citizens residing in a street (or in a portion of it) considering them as point sources. We note that, in the extreme case of zones which are not densely populated, a source may coincide even with a single house. Such collection points are serviced by a set of (small) collection vehicles (CV, in the following), leaving from a depot at the beginning of the planning horizon and going back to the depot at the end of the planning horizon. Before going back to the depot, each CV needs to visit a SWTS to transfer its load to a (long-haul) transportation vehicle (TV, in the following). If there is enough time before the planning horizon ends, a CV may travel an additional leg to visit other collection zones. In turn, each TV is in charge of transporting the solid waste to it final destination, i.e., a landfill. We assume that the capacity of a TV is such that it can consolidate the loads of more than a single CV before visiting the landfill. Moreover, the landfill acts as a depot for the TVs, meaning that they leave from it at the beginning of the planning horizon and go back to the landfill at the end of the working day. In the fashion of De Rosa et al. (2002), we assume that the time for loading/unloading vehicles is negligible compared to travel times.

As mentioned in Section 1, the problem can be decomposed in two stages, which are however strictly intertwined.

During the first stage, we need to define the routes for the CVs and schedule their visits to the SWTS. In particular, a route for a CV is made up of a number of legs, which can be of two types: legs of type 1 (T1, in the following), and legs of type 2 (T2, in the following). In particular, T1 legs start from the depot, visit a number of collection points and end at a SWTS, where the load of the CV is transferred to a TV. On the other hand, T2 legs start from a SWTS, visit a number of collection points, and go back to the SWTS (which in principle can be different form the initial one) to transfer the load to a TV. We notice that both T1 and T2 legs should be such that they keep enough time to go back to the depot, if no other legs can be added to the route. In practice, a route of a CV is made up of a single T1 leg and a set (possibly empty) of T2 legs. Moreover, the sum of the waste demands of the collection zones of each leg cannot exceed the vehicles' capacity.

During the second stage, we need to determine the routes for the TVs and synchronize their visits to the SWTS with those of the CVs. A route of a TV starts from the landfill, visits one or more SWTS, and returns to the landfill, where the vehicle is emptied.

Fig. 1 shows a graphical representation of the solid waste management scheme studied in this paper.

In the following, we describe the solution approaches we employ to face both stages of the decision process. In particular, in Section 3 we solve stage 1 and present a mathematical formulation, as well as a constructive procedure, and a matheuristic combining both previous approaches. On the other hand, in Section 4 we illustrate a heuristic approach to solve the second stage.

3. Collection phase solution approaches

As described previously, we tackle the first stage of the whole decision process by means of a mathematical formulation, a constructive procedure, and a combination of the both of them.

3.1. A mathematical formulation

A first approach to define the collection routes is formulating the problem as a mathematical model to minimize the number of utilized collection vehicles, and solving it by means of an off-the-shelf black-box solver.

For this purpose, we denote the maximum route duration by L, the set containing the vehicles of the (homogeneous) CV fleet by K, the set of collection points by C, and the set of solid waste transfer stations by S. Moreover, I represents the set of all T1 legs, whereas J is the set of all T2 legs. We assume that the number of collection points is such that the sets of all T1 and T2 legs can be obtained by complete enumeration. When

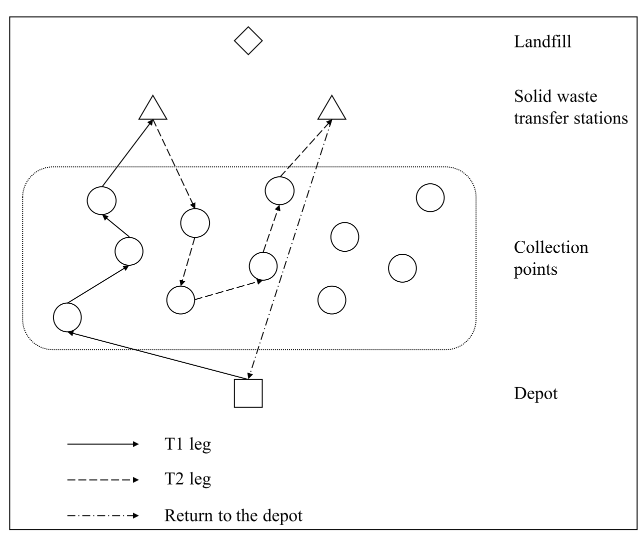

Fig. 1. Graphical representation of the solid waste management scheme.

enumerating the legs, we must ensure that the total leg time does not exceed the maximum route duration, and that the total amount of waste associated with the leg is lower than the vehicle capacity.

Before describing the optimization model, we need the to introduce some further notation. In particular: s_i and s_j denote the SWTS where T1 leg $i \in I$ ends, and the SWTS where T2 leg $j \in J$ starts, respectively; t_i^1 and t_j^2 indicate the time needed to complete T1 leg $i \in I$ and the time needed to complete T2 leg $j \in J$, respectively; t_j is the time needed to travel from the final SWTS of T2 leg $j \in J$ to the depot. In addition, a_{ci} is a binary parameter that takes value 1 if collection zone $c \in C$ is part of T1 leg $i \in I$ (0 otherwise), whereas b_{cj} is a binary parameter that takes value 1 if collection zone $c \in C$ is part of T2 leg $j \in J$ (0 otherwise).

The problem's decision variables are: x_i, binary variables that take value 1 if T1 leg $i \in I$ is selected, 0 otherwise; y_j, binary variables that take value 1 if T2 leg $j \in J$ is selected, 0 otherwise; v_{ki}, binary variables that take value 1 if T1 leg $i \in I$ is allocated to vehicle $k \in K$, 0 otherwise; w_{kj}, binary variables that take value 1 if T2 leg $j \in J$ is allocated to vehicle $k \in K$, 0 otherwise.

Thus, a mathematical model minimizing the number of utilized vehicles can be formulated as follows:

$$\min \sum_{k \in K, i \in I} v_{ki} \tag{1}$$

subject to

$$\sum_{i \in I} a_{ci} x_i + \sum_{j \in J} b_{cj} y_j = 1, \quad \forall c \in C \tag{2}$$

$$\sum_{i \in I} v_{ki} \leqslant 1, \quad \forall k \in K \tag{3}$$

$$\sum_{j \in J} w_{kj} \leqslant 1, \quad \forall k \in K \tag{4}$$

$$\sum_{k \in K} v_{ki} = x_i, \quad \forall i \in I \tag{5}$$

$$\sum_{k \in K} w_{kj} = y_j, \quad \forall j \in J \tag{6}$$

$$w_{kj} \leqslant \sum_{i \in I : s_i = s_j} v_{ki}, \quad \forall k \in K, j \in J \tag{7}$$

$$\sum_{i \in I} t_i^1 v_{kj} + \sum_{j \in J} (t_j^2 + t_j) w_{kj} \leqslant L, \quad \forall k \in K \tag{8}$$

$$x_i \in \{0, 1\}, \quad \forall i \in I \tag{9}$$

$$y_i \in \{0, 1\}, \quad \forall j \in J \tag{10}$$

$$v_{ki} \in \{0, 1\}, \quad \forall k \in K, i \in I \tag{11}$$

$$w_{kj} \in \{0, 1\}, \quad \forall k \in K, j \in J. \tag{12}$$

The objective function (1) aims to minimize the number of T1 legs, that is equivalent to minimizing the number of collection vehicles used. Constraints (2) impose that each collection zone is visited at most once, either during a T1 leg or during a T2 leg. Constraints (3) allocate at most one T1 leg to each collection vehicle (we assume there is no need to go back to the depot before the end of the working day). In addition, to keep the complexity of the model rather limited, we assume that at most one T2 leg can be allocated to each collection vehicle. This is achieved by means of constraints (4). Constraints (5) and (6) tie variables v_{ki} and w_{kj} to variables x_i and y_j, respectively, in such a way that each selected T1 leg and T2 leg is allocated to exactly one vehicle. Constraints (7) impose that a T2 leg can be allocated to a collection vehicle, if and only if a T1 leg has already been allocated to the same vehicle. In addition, the SWTS at which the T1 leg ends must be the same from which the T2 leg starts. Constraints (8) indicate that - for each vehicle - the total route duration, expressed as sum of the durations of the T1 and T2 legs, cannot exceed the maximum duration L. Finally, constraints (9)–(12) define the domain of the variables.

Formulation (1)–(12) results to be a Integer Linear Programming (ILP) model with binary variables. Solving it through a general purpose solver may be possible for small instances only, while solving realistic instances typically requires employing specific techniques or heuristic approaches. This is the purpose of Sections 3.2 and 3.3.

3.2. A constructive heuristic

As briefly mentioned in the previous section, when the instance size approaches that of a realistic instance, the complexity of the optimization model makes it very difficult to be solved optimally within a reasonable time by straightforwardly running a general purpose ILP solver. For this reason, in this section we propose a fast constructive heuristic approach, for which a pseudocode is reported in Algorithm 1.

Algorithm 1. A constructive heuristic to determine the routes of the collection vehicles

Input: the set C of collection zones, the waste quantity d_c of each $c \in C$, the vehicles capacity Q, the maximum route duration L, the set of SWTS S
Output: a set K of collection vehicles
1: **procedure** ConstructiveHeuristicCollectionRoutesC, d_c, Q, L, S
2: $K \leftarrow \varnothing$
3: $k \leftarrow 1$
4: **while** $C \neq \varnothing$ **do**
5: $R_k \leftarrow \{\text{depot}\}$ //route of vehicle k
6: $q_k \leftarrow Q$ //residual vehicle capacity
7: $T_k \leftarrow L$ //residual route time
8: **while true do**
9: $c' \leftarrow \mathit{ClosestCollectionZone}(\text{last}(R_k), C)$
10: $t' \leftarrow <$ time to visit c', a STWT, and the depot $>$
11: **if** $d_{c'} \leqslant q_k$ **and** $t' \leqslant T_k$ **then**
12: $R_k \leftarrow R_k \cup \{c'\}$
13: $q_k \leftarrow q_k \setminus d_{c'}$
14: $<$ update T_k $>$
15: $C \leftarrow C \setminus \{c'\}$
16: **else**
17: **if** $t' \leqslant T_k$ **then**
18: $s' \leftarrow \mathit{ClosestSWTS}(c', S)$
19: $R_k \leftarrow R_k \cup \{s'\}$
20: $q_k \leftarrow Q$
21: $<$ update T_k $>$
22: **else**
23: **break**
24: **end if**
25: **end if**
26: **end while**
27: **if** $\text{last}(R_k) \notin S$ **then**
28: $s' \leftarrow \mathit{ClosestSWTS}(\text{last}(R_k), S)$
29: $R_k \leftarrow R_k \cup \{\text{depot}\}$
30: **else**
31: $R_k \leftarrow R_k \cup \{s'\} \cup \{\text{depot}\}$
32: **end if**
33: $K \leftarrow K \cup \{R_k\}$
34: $k \leftarrow k + 1$
35: **end while**
36: **return** K
37: **end procedure**

The basic idea is to construct the vehicle routes by adding one collection zone at a time. In particular, until the set C becomes empty (line 4), we initialize a new route by starting from the depot, with residual vehicle capacity and route duration set as the maximum possible values (lines 5 –7). Then, we keep adding collection zones and solid waste transfer stations until this is feasible. More specifically, we iteratively extract a collection zone c' from set C as the closest to the last route point (line 9). Subsequently, if the insertion of c' into the current route does not violate both capacity and route duration constraints (line

11), we insert the collection zone into the current route, remove it from C, and update the residual capacity and residual time (lines 12 –15). If the insertion is not feasible, we check whether the infeasibility is due to the violation of the capacity constraint. If this is the case (line 17), we select the SWTS s' closest to the last route point, insert it into the current route, and update the residual capacity and residual time (lines 18 –21). On the other hand, if the insertion of c' is not possible because of the residual route duration (line 22), we stop adding zones to the current route (line 23). After checking whether the last route point is a SWTS or not, we consequently terminate the route by adding to it both the closest SWTS and the depot (lines 28 –29), or just the depot (line 31). Finally, we add the current route to the set K of collection routes and keep creating new routes until each collection zone is assigned to a vehicle.

3.3. A matheuristic

In this section, we describe a matheuristic - based on the destroy-and-repair para-digm (Pisinger & Ropke, 2010) - to obtain a solution for the first stage of our waste collection and transportation problem. In particular, we combine the constructive heuristic of Section 3.2 with the usage of the optimization model (1)–(12) on a reduced set of collection zones. The pseudocode of the resulting matheuristic is presented in Algorithm 2.

Algorithm 2. A matheuristic to determine the routes of the collection vehicles

Input: the set C of collection zones, the waste quantity d_c of each $c \in C$, the vehicles capacity Q, the maximum route duration L, the set S of SWTS, the number m of routes to be destroyed
Output: a set K of collection vehicles
1: **procedure** MatheuristicCollectionRoutesC, d_c, Q, L, S, m
2: $K \leftarrow$ *ConstructiveHeuristicCollectionRoutes*(C, d_c, Q, L, S)
3: $K' \leftarrow K$
4: new_routes $\leftarrow 0$
5: **while** new_routes $< m$ **do**
6: $<$ determine the route $k^* \in K'$ with the least capacity utilization $>$
7: $<$ calculate the center of gravity of each route in K' $>$
8: $K'' \leftarrow$ $<$ obtain the $(m-1)$ routes closest to k^* w.r.t. the center of gravity $>$
9: $K' \leftarrow K' \setminus \{k^*\} \setminus K''$
10: $C' \leftarrow$ $<$ extract the zones serviced by k^* and by the vehicles in K'' $>$
11: $K_{opt} \leftarrow$ $<$ solve model (1)–(12) using C' as set of collection zones $>$
12: **if** $K_{opt} \neq$ **null and** card$(K_{opt}) < h$ **then**
13: $K' \leftarrow K' \cup K_{opt}$
14: $K \leftarrow K'$
15: new_routes $\leftarrow$ card(K_{opt})
16: **else**
17: **break**
18: **end if**
19: **end while**
20: **return** K
21: **end procedure**

The procedure starts with the determination of an initial solution using the constructive heuristic (line 2). Then, we try to improve this solution by repeatedly removing m routes from the set of routes, and try to reassign the associated collection zones to a (possibly) lower number of vehicles (while loop at lines 5 –19). In particular, at each iteration we identify the route k^* having the least utilization in terms of vehicle capacity (line 6), we determine the center of gravity of each route (line 7), and then use such values to select a set K'' made up of the $(m-1)$ routes that are the closest to k^* in terms of them (line 8). The center of gravity of a route is straightforwardly obtained as the mean position of all the route points with respect to their coordinates in the plane. After removing route k^* as well the routes in K'' from the route set and determining the set C' of collection zones that were previously serviced by such routes (lines 9 –10), we use such set C' as (reduced) input for the mathematical model (1)–(12) (line 11). If the model provides a feasible solution and the number of routes is lower than m, we put together the new routes and those that were not removed previously (lines 12 –15), and iterate the destroy-and-repair phase. Otherwise, we stop the improving phase and output the best solution found (line 35).

4. Transportation phase solution approach

The goal of this phase is to determine the routes for a set of transportation vehicles, which must be synchronized with the visits of the collection vehicles at the SWTS. As described in Section 2, a route of a transportation vehicle starts from the landfill, visits one or more SWTS (where it collects the waste of one or more collection vehicles), and returns to the landfill (where the vehicle is emptied).

The routes of the collection vehicles are obtained as a result of using one of the approaches described in Section 3. Such routes determine a sequence of visits to the solid waste transfer stations, which in turn can be seen as a set H of tasks for the transportation vehicles. Each of these tasks $h \in H$ is characterized by: D_h, the overall quantity of waste carried by the collection vehicle at the SWTS; s_h, the SWTS visited by the collection vehicle, τ_h: the time instant at which the collection vehicle visits the SWTS. Thus, a task $h \in H$ can be represented as a triple (D_h, s_h, τ_h).

The constructive heuristic described in this section aims to sequence the tasks of set H, so to obtain the routes for a set E of transportation vehicles, for which the corresponding durations cannot exceed a threshold L'. Before describing the constructive heuristic - assuming that there is no waiting or queuing time at the SWTS - we need to introduce the following conditions that must hold in order for a task h to be feasibly inserted into a route R_e of a vehicle $e \in E$: (a) if $R_e \neq \emptyset$ and h' is the last task assigned to e, then the time needed to travel from $s_{h'}$ to s_h must be lower than or equal to the difference $\tau_h - \tau_{h'}$; (b) the residual capacity q_e of vehicle e must be greater than or equal to the overall quantity of waste D_h; (c) the duration of route R_e after the insertion of task h (and considering the time needed to travel back to the landfill) cannot exceed L'.

Algorithm 3. A constructive heuristic to determine the routes of the transportation vehicles

Input: the set H of tasks, the vehicles capacity Q', the maximum route duration L'
Output: a set E of transportation vehicles
procedure ConstructiveHeuristicTransportationRoutesH, Q', L'
$H \leftarrow$ $<$ sort the tasks so that $\tau_h \leqslant \tau_{h'}, \forall h \leqslant h'$ $>$
$E \leftarrow \emptyset$
$q_{min} \leftarrow \min\{D_h : h \in H\}$ //the lowest amount of waste associated with a task
while $H \neq \emptyset$ **do**
$h \leftarrow$ first(H)
$H \leftarrow H \setminus \{h\}$
$e \leftarrow$ *ChooseVehicle*(E, h)
if $e ==$ **null then**
$e \leftarrow$ card$(E) + 1$
$R_e \leftarrow \{$landfill$\} \cup \{s_h\}$ //route of vehicle e
$q_e \leftarrow Q'$ //residual vehicle capacity
$T_e \leftarrow L'$ //residual route time
$E \leftarrow E \cup \{e\}$
else
$R_e \leftarrow R_e \cup \{s_h\}$
$q_e \leftarrow q_e - D_h$
$<$ update T_e $>$
if $q_e < q_{min}$ **then**
$R_e \leftarrow R_e \cup \{$landfill$\}$
$q_e \leftarrow Q'$
end if
end if
end while
for e **in** E **do**
if last$(R_e) \neq \{$landfill$\}$ **then**
$R_e \leftarrow R_e \cup \{$landfill$\}$
end if
end for
return E
end procedure

A pseudocode of our approach is depicted in Algorithm 3. We first sort the tasks according to the values of τ_h (line 2). Then, we iteratively select and extract a task h, chosen as the first element of the task set H (lines 6 –7), and identify the best transportation vehicle to which to assign h (line 8). For this purpose, we make use of a cheapest-insertion strategy among all the vehicles for which conditions (a)–(c) are met (the details of this procedure are illustrated by means of the pseudocode of Algorithm 4). If such a vehicle does not exist (line 9), we initialize a new vehicle, by setting its partial route, its residual capacity, and its residual route time (lines 10 –13) and we add it to set E (line 14). On the other hand, if a vehicle is successfully identified, we add the SWTS s_h to its route, and update its residual capacity and route time (lines 16 –18). If, after this insertion, the residual capacity does not allow any additional insertions, we add a visit to the landfill and reinitialize the residual capacity to Q' (lines 19 –21). Finally, we check whether for some routes a visit to the landfill must be added and return the set E (lines 25 –30).

Algorithm 4. The procedure used to determine the vehicle to which to allocate a task

```
procedure ChooseVehicleE,h
  e←null
  bestInsertionCost ← +∞
  for e′ in E do
    cost ← < compute the best insertion such that (a)–(c) are met >
    if cost < bestInsertionCost then
      e←e′
      bestInsertionCost ← cost
    end if
  end for
  return e
end procedure
```

5. Computational results

In this section, we describe the experimental setting we use to assess our approaches, as well as the results of our tests. The mathematical formulation (1)–(12) is modeled using OPL and its solution is obtained employing IBM ILOG CPLEX 12.8.0 in its default mode and with all the standard cuts active. On the other hand, the heuristics of Sections 3.2, 3.3, and 4 are implemented in Java. All the tests are performed on a machine with an Intel processor clocked at 2.6 GHz and equipped with 8 GB of RAM.

The instances used to test our procedures are randomly generated, and are available at <http://www.emanuelemanni.unisalento.it/Sito/Test_Instances.html>. In particular, we assume that the service territory is modeled as a square with a side of 20 km, and the collection zones are such that both their coordinates are uniformly distributed in the interval $[0, 20]$. The amount of waste to be collected at each zone (in kilograms) is generated following a uniform distribution in $[100, 300]$, and the capacity of the vehicles is $Q = 400$ kg for the collection vehicles and $Q' = 1400$ kg for the transportation vehicles. With respect to the maximum duration of a route, we assume a duration of 480 min for both the collection and the transportation vehicles, i.e., $L = L' = 480$. Moreover, we consider that: (i) the collection vehicles depart from a central depot; (ii) there is a single landfill where the transportation vehicles are based; (iii) there are two solid waste transfer stations. Finally, concerning the travel times, they are obtained using the Euclidean distance and a fixed speed (30 km/h in our experiments).

Following the above setting, we have generated three groups of instances, depending on the number of collection zones. More specifically, we have: (1) small instances, with set C containing 20 or 40 zones; (2) medium instances, for which the number of zones varies in the set $\{60, 80, 100\}$; (3) large-scale instances, with the number of collection zones equal to 200, 400, 600, 800, or 1,000. For each group of instances and for each number of collection zones we have generated 20 instances.

The goal of our computational experiments is to compare the results obtained in three different situations. First, we consider the case in which the first stage is solved using the mathematical model with a time limit of 3,600 s, whereas the second stage is solved with the heuristic described in Section 4. Second, the first stage is solved using the constructive heuristic, whereas the second stage is solved with the heuristic described in Section 4. Third, we solve the first stage using the matheuristic, whereas the second stage is solved with the heuristic described in Section 4.

The comparisons are made in terms of number of collection and transportation vehicles needed and are reported in Tables 1–3, in which the table headings are: “zones”, number of collection zones; “opt. gap”, average optimality gap, as reported by the ILP solver at the end of the time limit, when solving the mathematical model; “CV”, number of collection vehicles needed for the first phase; “ρ CV”, average utilization rate of a collection vehicle, expressed as the fraction of its capacity needed to service all the collection zones assigned to it in a single leg; “TV”, number of transportation vehicles needed for the second phase. Moreover, we also compute the percentage deviation of: the number of collection vehicles, the average utilization of collection vehicles, and the number of transportation vehicles. These statistics are reported in the columns with headings “DEV CV”, “DEV ρ CV”, and “DEV TV”, respectively. The deviations of Table 1 are computed with respect to the exact approach, whereas those of Tables 2 and 3 are obtained by considering the constructive heuristic as benchmark. Each deviation is calculated as the difference between the result of the compared heuristic and the result of the benchmark, divided by the result of the benchmark. We emphasize that the aim of our computational experiments is to assess the (possible) reduction in the number of vehicles needed to perform the service and not to conduct an economic analysis related to such a reduction.

In the tables, each row reports the average result over the 20 instances. The detailed results obtained for each instance of each dataset are available at <http://www.emanuelemanni.unisalento.it/Sito/Test_Instances.html>.

First of all, it is worth noting that a straightforward usage of the mathematical model becomes not viable when the number of zones is greater than 40, since the solver is not able to obtain a feasible solution before the imposed time limit for such instances. For this reason, such results are reported in Table 1 only. As the table shows, even when the solver is able to find at least a feasible solution, the quality is not very good, with an average optimality gap of 99.99% and 73.33% when the number of zones is 20 and 40, respectively. For the small instances the constructive heuristic provides contrasting results when compared with the exact approach. Indeed, for the case with 20 zones, the former approach performs better than the latter, resulting in one collection vehicle less (a 13% reduction). On the other hand, when the zones are 40 the constructive heuristic performs worse that the mathematical model, needing five more vehicles to service all the zones (in percentage, the approach using the mathematical model for the first stage needs 42.50% less collection vehicles). When comparing the matheuristic and the exact approach, the former almost halves the number of required collection vehicles (a reduction of 44.50%) when considering 20 zones, with this gap dropping to 4.17% on the average when we consider 40 zones. With respect to the utilization rates, the average utilization is 0.60 for both zone sizes when the constructive heuristic is used, whereas comparing the exact approach and the matheuristic, the value of ρ is about the same with 40 zones, while for the instances with 20 zones the matheuristic outperforms the mathematical model (0.94 versus 0.62). Finally, concerning the output of the second stage (i.e., the number of transportation vehicles), we observe that there are not significant differences for the three approaches.

Considering the medium instances, the results of Table 2 show the matheuristic outperforming the constructive heuristic, with a reduction in the number of collection vehicles ranging from about eight vehicles

Table 1
Computational results for the small instances.

(a) Comparison between exact approach and constructive heuristic										
zones		Exact approach			Constructive heuristic					
	opt. gap	CV	ρ CV	TV	CV	DEV CV	ρ CV	DEV ρ CV	TV	DEV TV
20	99.99	10.00	0.62	2.05	8.70	−13.00%	0.60	−3.23%	2.00	−2.44%
40	73.33	12.00	0.96	3.90	17.10	42.50%	0.60	−37.36%	3.75	−3.85%
(b) Comparison between exact approach and matheuristic										
zones		Exact approach			Matheuristic					
	opt. gap	CV	ρ CV	TV	CV	DEV CV	ρ CV	DEV ρ CV	TV	DEV TV
20	99.99	10.00	0.62	2.05	5.55	−44.50%	0.94	52.51%	2.00	−2.44%
40	73.33	12.00	0.96	3.90	11.50	−4.17%	0.91	−4.86%	3.85	−1.28%

Table 2
Computational results for the medium instances.

zones	Constructive heuristic			Matheuristic					
	CV	ρ CV	TV	CV	DEV CV	ρ CV	DEV ρ CV	TV	DEV TV
60	25.10	0.61	5.20	17.05	−32.07%	0.92	50.23%	5.10	−1.92%
80	33.65	0.61	6.25	23.50	−30.16%	0.89	46.83%	6.15	−1.60%
100	41.65	0.61	7.50	29.75	−28.57%	0.88	44.37%	7.50	0.00%

Table 3
Computational results for the large-scale instances.

zones	Constructive heuristic			Matheuristic					
	CV	ρ CV	TV	CV	DEV CV	ρ CV	DEV ρ CV	TV	DEV TV
200	83.05	0.61	14.05	62.90	−24.26%	0.85	39.24%	13.70	−2.49%
400	164.65	0.61	26.20	125.60	−23.72%	0.83	35.54%	26.60	1.53%
600	247.30	0.61	37.85	191.85	−22.42%	0.81	33.09%	39.20	3.57%
800	326.90	0.62	50.45	259.10	−20.74%	0.80	29.60%	50.20	−0.50%
1,000	409.20	0.61	62.35	333.10	−18.60%	0.78	26.28%	63.55	1.92%

for the case with 60 zones, to about 10 and 12 vehicles when the number of zones becomes 80 and 100, respectively. In percentage, this reduction is −32.07% with 60 zones, −30.16% with 80 zones, and −28.57% with 100 zones. The average utilization is always 0.61 when the constructive heuristic is used. On the other hand, when the matheuristic is used, the utilization rate consistently rises, with a minimum value of 0.88 for the case with 100 zones, and a maximum value of 0.92 for the instances with 60 zones. Again, the number of transportation vehicles needed is substantially the same for the two heuristics.

With respect to the results of the tests on the large-scale instances of

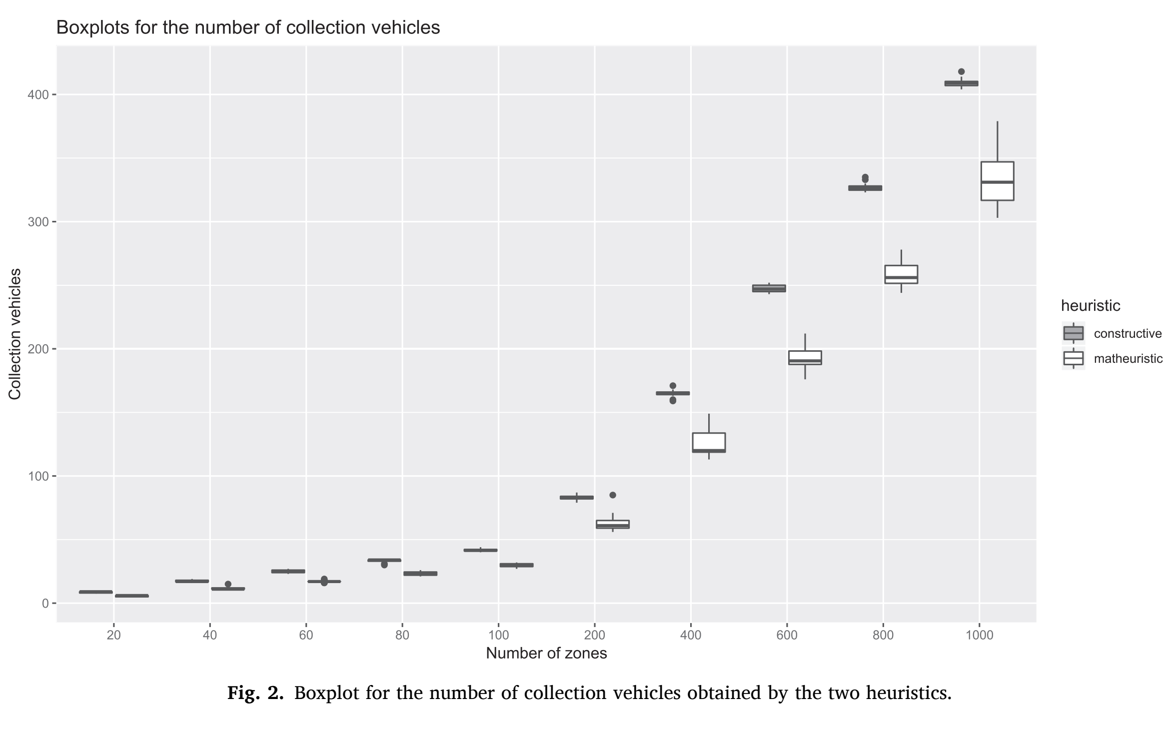

Fig. 2. Boxplot for the number of collection vehicles obtained by the two heuristics.

Table 3, they confirm the better performance of the matheuristic. In this case, the matheuristic allows to achieve a reduction in the number of collection vehicles ranging from about 20 for the case with 200 zones to about 76 for the largest instances. Considering the deviation, the highest value is observed for the instances with 200 zones (-24.26%), whereas the smallest value has been obtained for the dataset containing 1,000 zones (-18.60%). In addition, using the matheuristic allows to better utilize the collection vehicles, whose utilization rate is consistently higher than those of the constructive heuristic. In particular, the improvement of the value of ρ of the matheuristic ranges from a minimum of about 26% (1,000 zones) to a maximum of about 39% (200 zones). As for the previous cases, we do not observe any particular difference regarding the number of transportation vehicles. Indeed, both heuristic approaches yield to values that are almost the same for any tested number of collection zones.

Finally, we include a graphical analysis of the distribution of the number of collection vehicles utilized by the two heuristics for the 20 instances corresponding to each combination of parameters. The boxplots of such distributions (Fig. 2) confirm that the matheuristic outperforms the constructive heuristic, with the two boxplots not overlapping for any given number of zones. In most cases, the distributions are symmetric, with the constructive heuristic presenting a much lower dispersion than the matheuristic, as showed by the interquartile ranges. There are a few cases for which the matheuristic has a positive skewness, namely 400, 600, and 800 zones.

6. Conclusions

In this paper, we have considered a problem encountered in the municipal solid waste management sector, in which the waste is first collected by a fleet of small vehicles. Then, such vehicles visit a number of solid waste transfer stations, where the waste is transferred to larger transportation vehicles, which are in charge of transporting it to a landfill. In particular, we have studied the problem of determining the routes for both collection and transportation vehicles, as well as their synchronization at the transfer stations. More specifically, we have divided the problem into two phases and have devised several different approaches. For the collection phase, we have presented an exact approach utilizing a mathematical formulation, a constructive heuristic, and a matheuristic based on the destroy-and-repair paradigm. On the other hand, for the transportation phase we have proposed a heuristic approach. Computational results on a set of randomly-generated instances have shown that the approach combining the matheuristic for the collection phase with the heuristic for the transportation phase allows consistent reductions in terms of number of collection vehicles needed. In turn, such reductions can lead to substantial monetary savings for both the companies involved in the waste collection and transportation, and the citizens (in terms of taxes they have to pay for the service). Not less important, this can also yield a reduction in the environmental impact of the service, given that a lower number of vehicles can potentially travel over the streets.

References

Chu, F., Labadi, N., & Prins, C. (2006). A scatter search for the periodic capacitated arc routing problem. *European Journal of Operational Research, 169*, 586–605.

Das, S., & Bhattacharyya, B. K. (2015). Optimization of municipal solid waste collection and transportation routes. *Waste Management, 43*, 9–18.

De Oliveira, L. K., De Oliveira, R. L. M., & Rueda, L. S. T. D. A. (2017). Transshipment station for urban solid waste: an analysis considering city logistics concepts. *WIT Transactions on Ecology and the Environment, 223*, 559–569.

De Rosa, B., Improta, G., Ghiani, G., & Musmanno, R. (2002). The arc routing and scheduling problem with transshipment. *Transportation Science, 36*, 301–313.

Drexl, M. (2012). Synchronization in vehicle routing - a survey of vrps with multiple synchronization constraints. *Transportation Science, 46*, 297–316.

Eurostat statistics explained (2018). Municipal waste statistics. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php/Municipal_waste_statistics accessed on 2 December 2020.

Ghiani, G., Guerriero, E., Manni, A., Manni, E., & Potenza, A. (2013). Simultaneous personnel and vehicle shift scheduling in the waste management sector. *Waste Management, 33*, 1589–1594.

Ghiani, G., Guerriero, F., Laporte, G., & Musmanno, R. (2004). Tabu search heuristics for the arc routing problem with intermediate facilities under capacity and length restrictions. *Journal of Mathematical Modelling and Algorithms, 3*, 209–223.

Ghiani, G., Laganà, D., Manni, E., Musmanno, R., & Vigo, D. (2014). Operations research in solid waste management: a survey of strategic and tactical issues. *Computers & Operations Research, 44*, 22–32.

Ghiani, G., Laganà, D., Manni, E., & Triki, C. (2012). Capacitated location of collection sites in an urban waste management system. *Waste Management, 32*, 1291–1296.

Ghiani, G., Manni, A., Manni, E., & Toraldo, M. (2014). The impact of an efficient collection sites location on the zoning phase in municipal solid waste management. *Waste Management, 34*, 1949–1956.

Höke, M. C., & Yalcinkaya, S. (2021). Municipal solid waste transfer station planning through vehicle routing problem-based scenario analysis. *Waste Management & Research, 39*, 185–196.

Komilis, D. (2008). Conceptual modeling to optimize the haul and transfer of municipal solid waste. *Waste Management, 28*, 2355–2365.

Mostafayi Darmian, S., Moazzeni, S., & Hvattum, L. M. (2020). Multi-objective sustainable location-districting for the collection of municipal solid waste: Two case studies. *Computers & Industrial Engineering, 150*, 106965.

Pisinger, D., & Ropke, S. (2010). Large neighborhood search. In Handbook of metaheuristics (pp. 399–419). Springer.

Rízvanoğlu, O., Kaya, S., Ulukavak, M., & Yeşilnacar, M.İ. (2019). Optimization of municipal solid waste collection and transportation routes, through linear programming and geographic information system: a case study from Şanliurfa, Turkey. *Environmental Monitoring and Assessment, 192*.

Wang, G., Qin, L., Li, G., & Chen, L. (2009). Landfill site selection using spatial information technologies and AHP: a case study in Beijing, China. *Journal of Environmental Management, 90*, 2414–2421.

Willemse, E. J., & Joubert, J. W. (2016). Constructive heuristics for the mixed capacity arc routing problem under time restrictions with intermediate facilities. *Computers & Operations Research, 68*, 30–62.