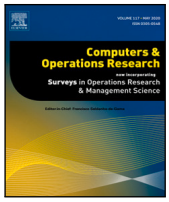
Computers & Operations Research 147 (2022) 105934

Contents lists available at ScienceDirect

Computers and Operations Research

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



Arc routing with trip-balancing and attractiveness measures — A waste collection case study

João Janela [a,b], Maria Cândida Mourão [a,c], Leonor Santiago Pinto [a,b,*]

[a] ISEG/UL - Lisbon School of Economics & Management, Universidade de Lisboa, Portugal
[b] REM - Research in Economics and Mathematics, CEMAPRE, Universidade de Lisboa, Portugal
[c] CMAFcIO, Universidade de Lisboa, Portugal

ARTICLE INFO

Keywords:
Mixed capacitated arc routing
Matheuristics
Mixed integer linear model
GIS
Attractiveness measure

ABSTRACT

This work focuses on a household (or door to door) waste collection problem in the Portuguese municipality of Seixal, which is modelled as a generalisation of a mixed capacitated arc routing problem (MCARP). The MCARP is known to be NP-hard. The proposed methodology uses: i) a GIS (geographic information system), for input/output and to reduce problem dimensions; ii) a matheuristic that iteratively solves a new hybrid model; and iii) two versions of a two-phase matheuristic. The latter pursues the generation of connected and compacted trips. During the first phase, called initial assignment, some links requiring service are assigned to vehicle services. In the second phase, the hybrid model finishes the assignment and generates a feasible set of trips. The quality of the generated solutions is assessed through the total time, as well as by some attractiveness measures. These measures evaluate the adequacy of the solutions for the real case-study, a crucial aspect for trips that need to be accepted by practitioners. With this purpose, a new attractiveness measure that introduces space dependent penalisation of overlaps, named weighted hull overlap, is also proposed. Computational results with 18 Seixal instances, with 265–1223 nodes and 492–2254 links, point to the good performance of the proposed methodology.

1. Introduction

Routing waste collection vehicles is a major concern for municipalities. The focus of this paper is a household waste collection problem in the Portuguese municipality of Seixal, in the Lisbon metropolitan area. To collect the refuse in a door-to-door system along the identified streets, each vehicle traverses a sequence of streets (collecting waste whenever needed), starting at the garage (or depot) and ending at the treatment facility. Both sides of some narrow streets can be collected (served) in just one zig-zag crossing. Each vehicle has a crew assigned to it with a corresponding labour contract. Thus, a time limit per day must be respected, which is represented through the vehicle time limit. According to practitioners, this time limit makes vehicle capacity redundant, and thus it need not be considered. In fact, when the time limit is reached, the vehicle's capacity is generally almost exhausted, but never attained, hence with no need to empty the vehicle during its service. Moreover, all vehicles need to be used, and, for fairness, the quantity of collected refuse (service) assigned to different vehicles must be similar, which calls for the services' balance. Defining a trip by a sequence of traversed links from the depot until the treatment facility, the problem under study aims to design balanced vehicle trips within the time limit, minimising the total time while including characteristics proposed by the municipality to make the solutions more adequate for practical implementation.

In several distribution or collection problems, the activities to be performed are spread along some predefined links (streets) of an associated undirected network, and the capacity of vehicles is relevant. Such cases are commonly modelled using the so-called capacitated arc routing problem (CARP). For a wide variety of applications that include two types of required links, directed and undirected, a mixed graph is involved. In this case, a natural option is the MCARP. The proof that MCARP is NP-hard follows from the fact that it generalises the CARP, which is known to be NP-hard (Golden and Wong, 1981). In books like Corberán and Laporte (2014) and Dror (2000), or in references such as Corberán et al. (2021), Corberán and Prins (2010), Golden et al. (2002), Mourão and Pinto (2017), and Sniezek et al. (2002), many real-world applications in the context of CARP or MCARP, can be found.

Arc routing problems (ARP) have been used to study household (or door-to-door) waste collection. Pioneering mixing ARP and GIS in a waste collection environment, Bodin et al. (1989) addressed the routing of sanitation vehicles in Oyster Bay, New York, USA. While GIS has

* Corresponding author at: ISEG/UL - Lisbon School of Economics & Management, Universidade de Lisboa, Portugal.
E-mail addresses: jjanela@iseg.ulisboa.pt (J. Janela), cmourao@iseg.ulisboa.pt (M.C. Mourão), lspinto@iseg.ulisboa.pt (L. Santiago Pinto).

https://doi.org/10.1016/j.cor.2022.105934
Received 4 November 2021; Received in revised form 27 June 2022; Accepted 28 June 2022
Available online 6 July 2022

often been used in the vehicle routing problem (VRP) this is not the case in ARP. Examples of waste collection ARP studies that use GIS include case studies in Philadelphia, Pennsylvania, USA (Sniezek et al., 2002) and Coimbra, Portugal (Santos et al., 2008). Malakahmad et al. (2014) used GIS to generate kerbside waste collection routes in Ipoh, Malaysia, and Ghiani et al. (2014) used it to produce instances for real waste collection networks. The latter tested their methodology on instances obtained from the Seixal municipality, Portugal, with 148 to 361 nodes and 284 to 713 links. However, the construction of their first trial data set suffers from some weaknesses, as it incorporates some estimates that could be improved with better automation.

Matheuristics have also been applied to routing models in waste collection systems, although they have been mainly developed to solve VRP (see Archetti and Speranza (2014)). We shall now refer to a few matheuristics that look at arc routing models in waste collection systems. Sniezek and Bodin (2006) proposed a composite approach which involves mixed integer programs, heuristics and a multi-criterion function to the routing of residential sanitation collection vehicles. The developed methodology, following the one in Sniezek et al. (2002), led to better results. Martins et al. (2013) and Ghiani et al. (2014) adapted the MCARP model of Gouveia et al. (2010) to the municipality of Seixal, Portugal. They considered the crews' working times and proposed two matheuristics. Braier et al. (2017) applied mathematical programming techniques to handle the recyclable waste collection problem in Morón, Buenos Aires, Argentina, which was seen as a particular case of a generalised directed open rural postman problem. The authors claim that the developed subtrips elimination algorithm allowed the generation of solutions in small computing times. In the periodic CARP (PCARP) paper of Chu et al. (2005) a two-phase heuristic starts by the clustering of tasks followed by the resolution of a single-period CARP per cluster. The methodology performance was assessed through adapted CARP benchmark instances. For the PCARP with irregular services, Monroy et al. (2013) developed a matheuristic that includes the resolution of a proposed model in the routing phase of a cluster-first/route-second approach. More recently, Zbib and Laporte (2020) proposed a data-driven matheuristic for a commodity-split multi-compartment CARP to address a real-life kerbside recyclable waste collection in Denmark. The problem, following one of Kiilerich and Wøhlk (2018), is defined on an undirected network, with a limited heterogeneous fleet of multi-compartment vehicles, with compartment capacities that can vary depending on the type of waste assigned to it and the respective compression factor in each vehicle type. The objective is to generate the set of least-cost routes from and back to the depot, collecting all the refuse types in each edge, within the capacities of each vehicle's compartments. A three-phase matheuristic was devised: assignment, routing, and selection phases. The computational results point to the advantage of combining different types of waste together in vehicles with more compartments. Finally, Tirkolaee et al. (2018) proposed a model for the multi-trip green CARP, considering environmental aspects. The model incorporates economic benefits, as well as the adverse impact of the CO_2 emissions in air pollution, and was validated in 15 randomly generated undirected instances with up to 214 required edges.

A common feature to all these matheuristics is that they provide a good compromise between heuristics and mathematical programming, taking advantage of mixed integer linear programming (MILP) solvers development.

These studies frequently discuss the adequacy of solutions to be implemented in practice. In fact, even in the first paper of Shuster and Schur (1974), it is stated that "routes should not be fragmented or overlapping. Each route should be compact, consisting of street segments clustered in the same geographical area". However, to our knowledge, very few studies propose measures to assess the attractiveness of ARP solutions (compact, connected and non overlapping routes).

Two studies that address these issues are Constantino et al. (2015) and Lum et al. (2017). Their proposed visual attractiveness measures are used to validate ARP solutions with multiple vehicles in terms of their aptness for practical implementation. More specifically, Constantino et al. (2015) propose three measures to assess the attractiveness of a given solution: the connectivity index (*CI*), the average task distance (*ATD*), and the routes overlapping index (*ROI*), while Lum et al. (2017) propose the hull overlap (*HO*). The actual definition and computation of these measures are addressed in Sections 5 and 7.3.

Corberán et al. (2017), when dealing with a min–max windy rural postman problem (RPP), incorporated some measures in their models, which they named aesthetic measures, by either adding new constraints or considering a multi-objective function. Clearly, assessing the attractiveness of the ARP solutions is something that warrants greater development.

This paper combines a generalisation of the MCARP, GIS, matheuristics and a new attractiveness measure to study the defined household waste collection problem. Feasible solutions are generated through the matheuristics, which resort to the use of a MILP formulation and a GIS (geography information system) to decrease the size of the MILP problems to solve.

Ideally, the vehicle trips are also balanced, connected and compact. We use the GIS available at the municipality, not only for the input/output phases, but also to decrease the problem's dimensions. As a result, the new data set here added is more realistic than that of Ghiani et al. (2014), with the input/output phases automatically controlled by QGis, a well-established free GIS software even though the municipality currently uses a proprietary software, *Arcview*. We also propose three matheuristics specifically developed for the case study. The first iteratively resorts to the resolution of a new hybrid model, and the other two are composed of two-phases. During the first phase, links requiring service, also named as tasks, are assigned to vehicles. This assignment uses GIS to promote vehicle services which may be considered attractive. The second phase calls for the resolution of the hybrid model to also identify the vehicle trips. One of these two-phase methods fixes all tasks to vehicles, defining zones, during the first phase, and thus only one trip per vehicle is generated in the second phase. In the other two-phase method, some tasks are released for the second phase. These remaining tasks are then assigned to vehicles within the generation of trips, through the hybrid model.

Finally, we propose a new measure to assess the attractiveness of the solutions, and use it together with other available measures.

Highlighting the major contributions of this paper, we stress:

- the development of a new hybrid model,
- the proposal of a new attractiveness measure,
- the use of a GIS environment that automatically generates the network, the corresponding data, and the solution presentation,
- the development of matheuristics that mix GIS, heuristic strategies and the hybrid mathematical model,
- the generation of appealing solutions to be implemented by practitioners.

Table 1 resumes the principal features of the referred papers, published after 2000, clarifying the main differences between them.

This paper is organised as follows. Section 2 describes the problem under study and presents the notation. The hybrid model is detailed in Section 3, before the description of the matheuristics, in Section 4. Section 5 is devoted to the new visual attractiveness measure, while Section 6, briefly refers to the use of GIS. Next, in Section 7, the computational results are presented and analysed, and Section 8 concludes the paper.

2. Problem description and notation

The collection network is described as a mixed graph $(N, A' \cup E)$. $A_R \subseteq A'$ and $E_R \subseteq E$ stand for the sets of required arcs and edges, respectively, representing the street segments with refuse to collect (tasks). $N = V \cup \{0', 0\}$ is the set of nodes, representing street crossings

Table 1
Referred papers in chronological order – CARP & Waste.

Paper	Methods	Case study & Networks	Highlights
Sniezek et al. (2002)	CH; GIS	Philadelphia, USA	Heterogeneous fleet; VSD & balanced
		MN; $\|A_R\|$ = 20,000 into 100 partitions (100 VSD)	Vehicle decomposition alg.: network partitioning & routing - RouteSmart GIS
Chu et al. (2005)	M; CH; Math	Adapted CARP instances 7-days horizon;	Periodic CARP: defined the no. of services per link in a planning horizon
		FS = 2–13 (1–4 per day); $\|E_R\|$ = 11–55	LP to define the problem & heuristics (two insertion & a 2-Phase) to solve it
Sniezek and Bodin (2006)	M;	4 VT;	CARP with HF with 4 VT & VSD
	Math	$\|A_R\|$ = 105–220; 95–98 trips; $\|A_R\|$ = 1776; 94–97 trips	Composite algorithm: model to generate a fleet mix; measure of goodness (costs, model compactness & overlapping) to better routes; Sniezek et al. (2002) improved
Santos et al. (2008)	CH; GIS	Coimbra, Portugal DN; $\|N_R\|$ = 20; 45 turn rules;	Demand links & nodes; TC & capacity; HF; traffic rules; multi-trips; depot≠DS
		FS = 5 (1–2 trips per vehicle); 1900 km of streets to serve	Spatial DSS: GIS with CH to generate data & trips; easy to perform what-if analysis and to be used by practitioners
Martins et al. (2013)	M; Math	Seixal, Portugal FS = 2–3; $\|A_R\|$ = 10–77;	MCARP with TC, also minimising the time difference between any two trips
		$\|E_R\|$ = 70–235	Some tasks are assigned to vehicles, from the solution of a relaxation, before solving the valid model
Monroy et al. (2013)	M; Math	14-days horizon (H); 3 arc classes; Instances adapted: $\|A_R\|$ = 22–194	Periodic CARP with irregular services & hierarchical link classes; respecting the passage frequencies per class and day
		(37–329 services in H); New random: $\|A_R\|$ = 14–68 (30–243 services in H)	Algorithms: cutting plane; cluster-1st/route –2nd, using a BP model for the clustering & MILP for the routing
Ghiani et al. (2014)	M; Math;	Seixal, Portugal	MCARP with TC & demand balanced trips
	GIS	FS = 2–5; $\|A_R\|$ = 43–99; $\|E_R\|$ = 70–291	GIS to generate real based instances; Based on Martins et al. (2013) with a new heuristic to assign tasks to vehicles
Malakahmad et al. (2014)	GIS	Malaysia: 5 routes in Buntong,	Kerbside waste collection
		Ipoh North, Kg Baru, Falim, Menglembu, $\|A_R\|$ = 15–91	GIS-ArcView; routes generation with Arc--GIS-NA & road conditions and topography
Constantino et al. (2015)	M; Math	CARP instances (mval, lpr) mval: FS = 4–12; A_R = 13–44;	Bounded overlapping MCARP, minimising routes overlapping
		E_R=28–106; lpr: FS = 3–25; A_R = 0–387; E_R = 11–748;	Generation of more attractive solutions with a small increase in UB values; attractiveness proposed measures: *CI*, *ATD*, *ROI*
Braier et al. (2017)	M; Math	Morón, Argentina	Open rural postman with traffic rules
		$\|A_R\|$ = 17–226; $\|E_R\|$ = 100–658	Subtrips elimination algorithm resulting in small computation times & better solutions
Lum et al. (2017)	CH	Real networks: Finland, UK, Australasia, Turkey, USA,	Min–max, windy multi-RPP, minimising the cost of the longest route
		$\|E\|$ = 586–1540; FS = 3,5,10 Random: $\|E\|$ = 420–1112	Heuristic with compactness and overlapping proposed measure: *HO*
Corberán et al. (2017)	M; CH;	Instances with FS = 2–4:	Multi-objective, min–max, windy multi-RPP
	B&C	$\|E_R\|$ = 24–65; $\|E_R\|$ = 40–60; Paris $\|E_R\|$ = 44; San-Francisco $\|E_R\|$ = 29 With FS = 2–5: $\|E_R\|$ = 100–400	Several models & a heuristic that extends the one of Lum et al. (2017) by including attractiveness measures in the improvement phase and a perturbation routine
Kiilerich and Wøhlk (2018)	M	Denmark: 6 counties Large instances proposed for the CARP variants	≠wastes; HF with multi-compartment vehicles coordinated; periods; models for 5 CARP variants; no computational results
Tirkolaee et al. (2018)	M; CH; SA; GA	$\|E_R\|$ = 6–214; 2 VT: FS = 1–13; FS = 1–11	Minimise CO_2 emissions; multi-trips; depot≠DS; TC & capacity
			Model vs. Hybrid Genetic Alg. (CH; SA; GA)

(*continued on next page*)

Table 1 (*continued*).

Paper	Methods	Case study & Networks	Highlights
Zbib and Laporte (2020)	M; Math	Denmark: 6 counties $\|E_R\|$ = 19–3797; 4–6 VT	≠wastes; HF with multi-compartment vehicles commodity split; 3 decision levels
		(1–4 compartments); FS = 16–160	Data-driven 3-phase matheuristic
This work	M; Math;	Seixal, Portugal	General-CARP; depot≠DS; TC; balanced trips
	GIS	FS = 2–6; $\|A_R\|$ = 119–503; $\|E_R\|$ = 138–520	Hybrid model; 3 matheuristics: hybrid; 2-phase (cluster-1st/route-2nd); 2-phase variant; Attractiveness measure

Legend: $|A_R|$-no of arc-tasks; B&C-branch&cut; BP-binary programming; CH-constructive heuristic; DN-directed network; DS-dump site; DSS-decision support system; $|E|$-no of edges; $|E_R|$-no of edge-tasks; FS-fleet size; GA-genetic algorithm; HF-heterogeneous fleet; LP-linear programming; M-model; Math-matheuristic; MILP-mixed integer LP; MN-mixed network; $|N_R|$-no of demand nodes; SA-simulated annealing; TC-time constrained; UB-upper bound; VSD-vehicle site dependencies; VT- no of vehicle types.

or dead-end streets (set V), the depot, node $0'$, and the treatment facility, node 0. No tasks are incident either in the depot or in the treatment facility.

A directed graph $G = (N, A)$ is obtained by replacing each edge $e \in E$ with two opposite linked arcs, i.e. $A = A' \cup \{(i,j),(j,i) : (i,j) \in E\}$. If the edge $e \in E_R$ is an edge-task, the edge is considered to be serviced when one of its two corresponding arcs is chosen to be serviced.

The set of required arcs is denoted by $R \subseteq A$, with cardinality $|R| = |A_R| + 2|E_R|$. Note that the number of arcs that will be serviced remains equal to $|A_R| + |E_R|$, as edges in E_R are served in only one direction.

Each arc $a = (i,j) \in A$ has associated a deadheading time, d_a, that is the time needed to traverse it without being served, and if $a = (i,j) \in R$, q_a is its demand, representing the refuse to collect (service), and c_a ($c_a > d_a$) the corresponding service time.

The set of vehicles is represented by K, and $k \in K$ denotes a specific vehicle. The service to be performed by each vehicle (waste to be collected) is identified and assigned to a specific designed collecting zone. The refuse in each zone must be collected by a vehicle crew in only one trip, starting at the depot and ending at the treatment facility, and respecting the time limit, L. This value, L, represents the crew working time minus the dump time at the treatment facility and the time to return to the depot after emptying the vehicle. Again, in this setting, vehicle capacity is redundant.

As usual, sets $\delta_X^+(i)$ and $\delta_X^-(i)$ are used to identify all arcs, in X, with node i as an initial and final node, respectively.

A MILP model that will be used in the methodology to find feasible solutions to the problem defined is formulated in the next section.

3. Hybrid formulation

Gouveia et al. (2010) presented a compact formulation for the MCARP, based on flow variables, and an aggregation over the vehicles, i.e. relaxing the capacity constraints and thus only generating one giant trip. Both the valid and the aggregated models have the same linear relaxation bound. These authors claim that the valid model provides optimal solutions for medium sized instances and good lower bounds on larger instances. Moreover, the aggregated model generates good lower bounds in small computation times, even for the larger instances.

The aim of this paper is to generate feasible solutions by exploring a combination of these two models. The main point is then to derive a hybrid formulation that forces the identification of a chosen number of valid trips, α, i.e. within the time limit, L, and generates only one giant trip including all the remaining tasks. Note that this giant trip can be much smaller than the one generated by the aggregate model of Gouveia et al. (2010).

The time allowed for this giant trip equals L times the number of trips aggregated in it. Set $P = \{p : p = 0, \dots, \alpha\}$, stands for the set of trips, where $p = 0$ refers to the giant trip and the remaining $\alpha < |K|$ trips are the valid ones.

The motivation behind the hybrid model is to be able to solve bigger instances using the methodology detailed in the next section.

Although suggested from the models in Gouveia et al. (2010), we stress that, following the reality at the municipality of Seixal, the model here presented aims to identify trips that are time limited instead of considering the vehicle capacity.

In addition, a minimum number of tasks or a minimum service per trip is also imposed to balance the service among crews. The remaining attractiveness features desired for zones are previously tackled in the first phase of the two-phase matheuristics, and thus not enforced by the model.

Three sets of variables are used: (i) for each task $a \in R$, a set of binary variables x_a^p identifies the trip p that will be used to collect the refuse on street a; (ii) for each arc $a \in A$, integer variables y_a^p count the number of times arc $a = (i,j)$ is deadheaded during trip p; and (iii) f_a^p is the flow related with the remaining time to finish trip p. To simplify the presentation of the model, we assume that $x_a^p = 0, \forall a \in A \setminus R$.

The hybrid model (HM) is defined as follows.

$$\min \sum_{p=0}^{\alpha} \sum_{a \in A} (d_a y_a^p) \tag{1}$$

subject to:

$$\sum_{j \in \delta_A^+(0')} y_{0'j}^0 = |K| - \alpha \tag{2}$$

$$\sum_{j \in \delta_A^+(0')} y_{0'j}^p = 1 \qquad p = 1, \dots, \alpha \tag{3}$$

$$\sum_{j \in \delta_A^-(i)} f_{ji}^p - \sum_{j \in \delta_A^+(i)} f_{ij}^p = \sum_{j \in \delta_R^-(i)} c_{ji} x_{ji}^p + \sum_{j \in \delta_A^-(i)} d_{ji} y_{ji}^p \qquad i \in V, p \in P \tag{4}$$

$$\sum_{j \in \delta_A^+(0')} f_{0'j}^p = \sum_{a \in R} c_a x_a^p + \sum_{a \in A} d_a y_a^p \qquad p \in P \tag{5}$$

$$\sum_{i \in \delta_A^-(0)} f_{i0}^p = \sum_{i \in \delta_A^-(0)} d_{i0} y_{i0}^p \qquad p \in P \tag{6}$$

$$f_a^p \leqslant L \left(y_a^p + x_a^p\right) \qquad a \in A, p = 1, \dots, \alpha \tag{7}$$

$$f_a^0 \leqslant L\,(|K| - \alpha)\left(y_a^0 + x_a^0\right) \qquad a \in A \tag{8}$$

$$\sum_{p=0}^{\alpha} x_{ij}^p = 1 \qquad (i,j) \in A_R \tag{9}$$

$$\sum_{p=0}^{\alpha} \left(x_{ij}^p + x_{ji}^p\right) = 1 \qquad (i,j) \in E_R : i < j \tag{10}$$

$$\sum_{j \in \delta_A^+(i)} y_{ij}^p + \sum_{j \in \delta_R^+(i)} x_{ij}^p - \sum_{j \in \delta_A^-(i)} y_{ji}^p - \sum_{j \in \delta_R^-(i)} x_{ji}^p = 0 \qquad i \in V, p \in P \tag{11}$$

$$\sum_{j \in \delta_A^+(0')} y_{0'j}^p = \sum_{i \in \delta_A^-(0)} y_{i0}^p \qquad p \in P \tag{12}$$

$$f_a^p \geq c_a x_a^p + d_a (y_a^p - 1) \qquad a \in A; p = 1, \dots, \alpha \tag{13}$$

$$f_a^p \geq c_a x_a^p \qquad a \in R; p = 1, \dots, \alpha \tag{14}$$

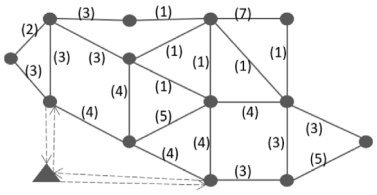


Fig. 1. Network to apply the matheuristics.

$$f_a^0 \geq c_a x_a^0 + d_a(y_a^0 - |K| + \alpha) \qquad a \in A \qquad (15)$$

$$f_a^0 \geq c_a x_a^0 \qquad a \in R \qquad (16)$$

$$x_a^p \in \{0,1\} \qquad a \in R, p \in P \qquad (17)$$

$$y_a^p \in \mathbb{N}_0; f_a^p \geq 0 \qquad a \in A, p \in P \qquad (18)$$

More precisely, the problem formulated is to find a set of $\alpha + 1$ vehicle trips (imposed by Constraints (2) and (3)), of which α satisfy the vehicles' time limit of L (Constraints (3)–(7)), and a time limit of $(|K|-\alpha)L$ for the giant trip ($p = 0$) (Constraints (2), (5) and (8)), starting at the depot, node $0'$ (by Constraints (2) and (3)), and servicing all the required links (enforced by Constraints (9) and (10)), with total minimum time (1). Note that Constraints (10) are also used to define the direction in which to serve an edge-task. Isolated subtrips, i.e. trips not including nodes 0 and $0'$, are prevented through the use of the flow variables in Constraints (2)–(8). The balance at each node is forced by Constraints (11), and Constraints (12) impose that each vehicle leaves the depot the same number of times it arrives at the treatment facility. Although not needed, inequalities (13)–(16) are introduced to reinforce the model with the definition of some lower bound values on the flow variables.

Note that this hybrid model lies between the aggregated and the valid models from Gouveia et al. (2010). In fact, as α goes from 1 to $|K| - 1$ the hybrid model approaches the valid one, and only when $p = 0$ is considered (neither α nor the related constraints, (3), (7), (13), and (14) are included) does (HM) coincide with the aggregated model, which generates one giant trip only. Different values of α can then be used to generate different solutions, as will be seen in the matheuristics. Moreover, when $\alpha = |K| - 1$ the (HM) is synonymous with a MCARP where the time limit is replaced by the capacity. In that case, Constraints (2) and (3) ensure that all $|K|$ vehicles leave the depot. The flow variables and Constraints (4)–(8) impose the time limit of L per trip and that no isolated subtrips exist. Constraints (9)–(12) have the meaning above referred, guaranteeing the services in all tasks, the equilibrium of the nodes, as well as the domains of variables.

To enforce balanced trips two types of constraints were tested for a given parameter $\beta \in [0,1]$, namely:

$$\frac{m}{|K|}(1-\beta) \leq \sum_{a \in R} x_a^p \leq \frac{m}{|K|}(1+\beta) \quad p \in P \setminus \{0\} \qquad (19)$$

$$\frac{Q_T}{|K|}(1-\beta) \leq \sum_{a \in R} q_a x_a^p \leq \frac{Q_T}{|K|}(1+\beta) \quad p \in P \setminus \{0\} \qquad (20)$$

where $m = |A_R| + |E_R|$ is the number of tasks and $Q_T = \sum_{a \in E_R \cup A_R} q_a$ is the total demand.

While Constraints (19) assign a similar number of tasks to different vehicles, through Constraints (20) the balance refers to the demand to be collected by each vehicle.

4. Matheuristics

Feasible solutions are pursued by three matheuristics, one detailed in Algorithm 1 (H1) that sequentially solves the hybrid model, and a two–phase matheuristic as detailed in Algorithm 2 (H2). A variant of the former is also described in Algorithm 5 (H3). The idea is to compare methodologies supported by different reasoning. Although all use (HM), (H1) iteratively identifies one valid trip, while (H2) may be considered a cluster-first/route-second heuristic. (H3) releases some of the assignments to try to take advantage of completing the zoning while routing is performed. Note that even though for a particular network one of the heuristics may perform consistently better, this is not generally true.

Example 1. Fig. 1 depicts the network of a small example that is used to illustrate the matheuristics. We assume three vehicles, with a time limit of 60. For simplicity the depot and the treatment facility coincide. Deadheading times are equal to 2, except for the links incident in the depot/treatment facility (triangle node) which are equal to 5, as the service times. Each task demand is near the corresponding link in brackets.

4.1. Hybrid matheuristic (H1)

In short, in each iteration of the hybrid matheuristic the (HM) with $\alpha = 1$ is solved. Then, the valid trip that is identified is kept (the corresponding x variables are fixed to that vehicle trip) and the tasks it serves are considered as deadheading links in the network for the next iteration. It ends when the giant trip is also a feasible trip for just one vehicle or if no feasible solution is found. Indeed, this heuristic may end up without a set of feasible trips. This happens when the integer solver fails to find a feasible solution for the (HM), or when in the last iteration the giant trip is not a feasible one, i.e., it does not respect the time limit for one vehicle, as detailed in Algorithm 1.

Algorithm 1 – H1 (Hybrid Model Matheuristic)

Require: network; $|K|$
$Sol \leftarrow$ **true** ▷ **true** if (HM) found a solution
$Giant \leftarrow$ **false** ▷ **true** if the giant trip found by (HM) is feasible
$nt \leftarrow 0$ ▷ no. of feasible trips
$FT \leftarrow \emptyset$ ▷ set of feasible trips
while $(nt < |K| - 1) \wedge (Sol)$ **do**
  $Sol \leftarrow$ **false**
  $nk \leftarrow |K| - nt - 1$ ▷ no. of trips to consider in (HM)
  Solve (HM) with $\alpha = 1$ and replacing $|K|$ by nk
  Update Sol
  Update $Giant$
  if Sol **then**
    $FT \leftarrow FT \cup \{$feasible trip generated by (HM)$\}$
    $nt \leftarrow nt + 1$
    Fix $x_{ij}^{nt} \leftarrow 1$ according to the generated feasible trip
    if $Giant$ **then**
      $FT \leftarrow FT \cup \{$giant trip generated by (HM)$\}$
      $nt \leftarrow nt + 1$
    end if
  end if
end while
if $\neg(Giant) \vee \neg(Sol)$ **then**
  "The heuristic fails to find a feasible solution"
end if
Ensure: FT ▷ in case one feasible solution was found

Example 2. Considering Example 1, Fig. 2 presents the solution found by the Hybrid Matheuristic. Fig. 2(a) shows the output of the first iteration, that is: (i) one giant trip, with a total time of 85 (13 tasks and 4 deadheading arcs incident to the depot/treatment facility), greater than the time limit of 60 (black thick lines); and (ii) a valid identified trip with a total time of 59 (blue dashed lines, with 9 tasks, 2 deadheading arcs and the links incident in the depot/treatment facility). Note that, the giant trip, representing the aggregate service of two vehicles, has two arcs out of the depot and two arcs into the treatment facility (the same triangle node). Fig. 2(b) presents the final solution for the three vehicles, obtained at the end of the second iteration, spending 59, 40

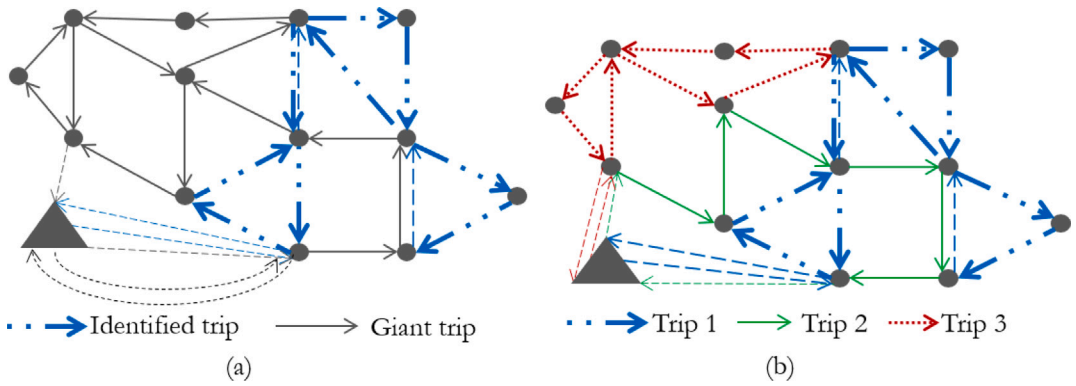


Fig. 2. Hybrid model matheuristic.

and 45, respectively, which correspond to a total deadheading time of 34.

4.2. Two-phase matheuristic (H2)

The two–phase matheuristic (Algorithm 2) first assigns tasks to vehicles (Algorithm 3) and then generates feasible trips (one per vehicle) through the use of the hybrid model (HM), thus falling under the classification of a cluster-first/route-second type heuristic.

Algorithm 2 – H2 (2–Phase Matheuristic)

Require: network; $|K|$
//1st–phase – assignment ▷ assign tasks to vehicles and layers
Call AssignZoneLayer() ▷ Algorithm 3
//2nd–phase – routing
Solve (HM) with $\alpha = |K| - 1$ ▷ identify $|K|$ feasible trips
Ensure: FT ▷ feasible solution

The first phase assigns tasks to each vehicle's service and then balances them. More specifically, vehicle services are augmented by layers of adjacent links, starting from a set of fixed $|K|$ core-tasks (arc or edge tasks). The core-tasks are fixed by practitioners, who can easily identify tasks that should belong to different zones as well as tasks that must belong to the same zone. Nevertheless, in order to allow a wider variety of solutions, core-tasks are fixed among those that, once identified as having to be in different zones, are far from each other, as next detailed.

The algorithm starts by randomly picking one task, from those identified, and this task represents the core of the first zone. Then, a task that must belong to a different zone and farther from the first core is chosen to be the core of the second zone. The third is the farthest from the previous two, and it must also be in a different zone, and so on and so forth. For this purpose, the distance between two tasks $a = (i, j)$ and $b = (u, v)$, D_{ab}, is the minimum among the minimum distances between the corresponding ending nodes, that is, $D_{ab} = \min\{D_{iu}, D_{iv}, D_{ju}, D_{jv}, D_{ui}, D_{vi}, D_{uj}, D_{vj}\}$. For instance, D_{iu} is the minimum distance from $i \in V$ to $u \in V$. Additionally, the distance from task, a, and two other tasks, b, c, is the sum of the correspondent distances, $D_{a,bc} = D_{a,b} + D_{a,c}$.

From the set of the $|K|$ chosen core-tasks, the aim is then to grow each zone around its corresponding core-task respecting the vehicle time limit, L. Each zone is first enlarged with all the links adjacent to its core, following an increasing order of current zones' demand.

The procedure is repeated, growing zones by layers, until every task is assigned. The core-task is assigned to layer 0, the core of the zone, and the layer index is increased by one each time a zone is enlarged. Algorithm 3 details this procedure.

In this context, each zone is divided into layers: the core layer and remaining layers, each containing the tasks that stand apart equally (in number of links) from its core. Thus, after the first stage of the first phase, the layer of a task corresponds to the number of links it is away from its zone core (core-task) plus one.

Algorithm 3 – AssignZoneLayer

Require: network; $|K|$; L
$ST \leftarrow$ set of $|K|$ core–tasks ▷ chosen from the ones identified by the practitioners
$SAL \leftarrow \emptyset$ ▷ set of assigned links
$k \leftarrow 1$ ▷ zone index
while $k \leq |K|$ **do**
  Randomly chose a core–task $s \in ST$
  $SAL \leftarrow SAL \cup \{s\}$
  $ST \leftarrow ST \setminus \{s\}$
  Set s to layer 0 of zone k
  $k \leftarrow k + 1$
  $Nl[k] \leftarrow 0$ ▷ no. of identified layers in zone k apart from the core
end while
// Define new layers until all tasks have a layer and a zone
while $SAL \neq R$ **do**
  Order zones by increasing demand $k_1, \cdots, k_{|K|}$
  for $i = 1, \cdots, |K|$ **do**
    Identify links in $A' \setminus SAL$ adjacent to a link in layer $Nl[k_i]$, within L
    Enlarge zone k_i with all identified links
    $Nl[k_i] \leftarrow Nl[k_i] + 1$
    Set all the identified tasks to layer $Nl[k_i]$ of zone k_i
    Update SAL with all identified tasks, and inverse tasks for edges, if any
  end for
end while
Call BalanceZones() ▷ Algorithm 4
Fix $x_{ij}^k \leftarrow 1$ accordingly the respective zone
Ensure: A zone and a layer for each task, and x_{ij}^k

A second stage of the first phase is then applied in order to balance the service of the vehicles (Algorithm 4), and thus tasks assigned to vehicles with larger assigned demand are moved to smaller neighbouring zones. These services are considered balanced when the percentage of increase in demand from the smallest zone to the largest one is less than 1%. This procedure is repeated a predefined maximum number of times ($MaxIt$) if balance is not achieved.

Example 3. Considering Example 1, Fig. 3 shows the three zones. The core of zone 1, 2, and 3 is marked as Z_1, Z_2, and Z_3, respectively. Values next to remaining edges represent the number of the correspondent assigned layer. After assigning layers and zones and to balance them,

Algorithm 4 – BalanceZones

Require: A zone and layer for each task; $MaxIt$ ▷ max no. of iterations
$Iterations \leftarrow 0$
$Stop \leftarrow$ **false**
$EQ \leftarrow$ **false**
while $\neg Stop$ **do**
  for $t = 1, \cdots, |SAL|$ **do**
    $nvehicle(t) \leftarrow$ zone index of task t
    $nlayer(t) \leftarrow$ layer index of task t
    Identify neighbours of t not in $nvehicle(t)$: $u_1, \cdots, u_{nei}$
    for $j = 1, \cdots, nei$ **do**
      $nvehicle(u_j) \leftarrow$ zone index of task u_j
      $nnlayer(u_j) \leftarrow$ layer index of task u_j
      if $demand(nvehicle(u_j)) > demand(nvehicle(t))$ **then**
        $nvehicle(u_j) \leftarrow nvehicle(t)$
        $nlayer(u_j) \leftarrow nlayer(t) + 1$
      end if
    end for
  end for
  $Iterations \leftarrow Iterations + 1$
  Verify and fix connectivity
  if $(\max(demand) - \min(demand))/\max(demand) < 1\%$ **then**
    $Stop \leftarrow$ **true**
    $EQ \leftarrow$ **true**
  end if
  if $Iterations \geq MaxIt$ **then**
    $Stop \leftarrow$ **true**
  end if
end while
if EQ **then**
  Update zones and layers
end if
Ensure: A zone and a layer for each task

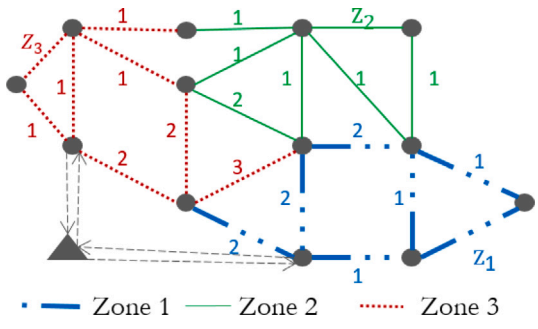


Fig. 3. 2-phase matheuristic — 1st phase output.

the 2nd-phase calls (HM) to generate the routing solutions for the three vehicles.

The second phase of the matheuristic (Algorithm 2) is accomplished by the solution of the (HM) with $\alpha = |K| - 1$ and considering the assignment of tasks resulting from the first phase. More specifically, if task (i, j) is assigned to zone k, the value of the correspondent variables must be fixed to one, that is, $x_{ij}^k = 1$ if $(i, j) \in A_R$ and $x_{ij}^k + x_{ji}^k = 1$, in case $(i, j) \in E_R$.

Note that, because every task is fixed, (HM) only determines the trips and the direction to service each edge task.

Alternatively, the assigned zone can be ignored for some tasks, based for instance on the task layer. In these cases, the (HM) also selects a vehicle to service those tasks, as next described in the two-phase matheuristic variant.

4.3. Two-phase matheuristic variant (H3)

Because it is a natural and easy to implement variant, this heuristic is developed in the search for a methodology capable of generating the best feasible solutions. The idea is to try to take advantage of some initial assignments, leaving the choice of remaining assignments to the (HM). This is justified for three main reasons. Firstly, by releasing only some tasks from the initial assignment we try to give some freedom to the model hoping that this may lead to better solutions. Secondly, since the solver fails to produce feasible solutions when no initial assignment is performed, the released tasks are chosen so that more feasible solutions may be attained. Finally, the generation of zones through Algorithm 3 may prematurely leave some zones without neighbours, thus preventing its enlargement. This drawback may hopefully be repaired if some assignments from the outer layers are realised. Thus, exploring the flexibility of (HM) to also perform part of the zoning is the purpose of the variant discussed in this section and detailed in Algorithm 5.

Algorithm 5 – H3 (2–Phase Matheuristic Variant)

Require: network; $|K|$; γ; $rule$
  // 1st–phase – assignment
Call AssignZoneLayer() ▷ Algorithm 3
if $rule = 1$ **then**
  for $k = 1, |K|$ **do** ▷ release some assignments according to rule 1
    $VehDemand \leftarrow$ Total demand in zone k
    $FracDemand \leftarrow \gamma \times SecDemand$
    while $VehDemand > FracDemand$ **do**
      Release tasks from the outer layer
      Update VehDemand and outer layer number
    end while
  end for
end if
if $rule = 2$ **then**
  for $k = 1, |K|$ **do** ▷ release some assignments according to rule 2
    while ∃ a task not yet analysed $(i, j) \in$ zone k **do**
      if (i, j) is in a layer $> \gamma$ **then**
        Release task (i, j)
      end if
    end while
  end for
end if
if $rule = 3$ **then**
  for $k = 1, |K|$ **do** ▷ release some assignments according to rule 3
    while ∃ a task not yet analysed $(i, j) \in$ zone k **do**
      if (i, j) is in a layer $> Nl[k] - \gamma$ **then**
        Release task (i, j)
      end if
    end while
  end for
end if
Set $x_{ij}^k \leftarrow 0$ for released tasks
  // 2nd–phase – routing and complete assignment
Solve (HM) with $\alpha = |K| - 1$ and the fixed x_{ij}^k values ▷ identify $|K|$ feasible trips
Ensure: FT ▷ feasible solution

Tasks are released from their initially proposed zone using three rules that take into account the task layer number. The first one is based on demand and releases from each zone a number of outer layers (all tasks) in such a way that a fraction γ_1 of the zone demand remains assigned. The second rule releases from all zones every task with a layer number greater than γ_2. The third rule releases from all zones every task in the last γ_3 layers. Note that even when $\gamma_2 = \gamma_3$ the two last rules may differ if zones have different numbers of layers, as next illustrated.

Example 4. Figs. 4(a), (b) and (c) show the tasks released from the three zones (Z_1, Z_2, and Z_3) of Example 3, according to rule $r = 1, 2, 3$,

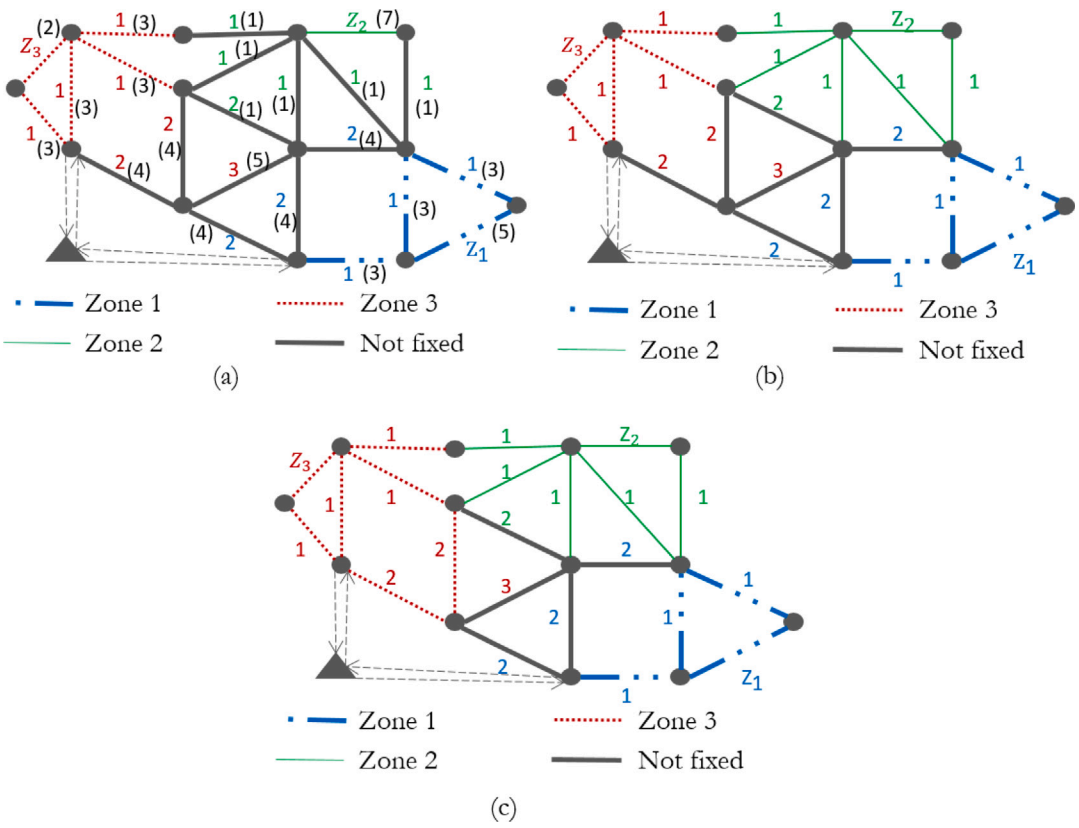


Fig. 4. 2-phase variant — 1st phase output for rule: (a) $r = 1$; (b) $r = 2$; (c) $r = 3$.

respectively. Here we used $\gamma_1 = 0.5$, $\gamma_2 = \gamma_3 = 1$. Further discussion on the choice of these parameters can be found in Section 7. As before, cores (layer 0) are marked with the respective zone index. Assuming task demands, given in brackets near the links in Fig. 4(a), keep 0.5 of each zone demand fixed corresponds to fix $0.5(5+9+12) = 13, 0.5(7+5+1) = 6.5$, and $0.5(2+12+8+5) = 13.5$, in zones 1, 2, and 3, respectively. Thus, layers to keep fixed are 0 and 1 in zone 1 (with demand equal to $14 > 13$), 0 in zone 2 (demand equal to $7 > 6.5$), and layers 0 and 1 in zone 3 (demand equal to $2 + 12 > 13.5$). All tasks in a layer are kept fixed from the core layer until the first layer which makes the total fixed demand greater than γ_1. In Fig. 4(b), according to rule 2, all tasks in layers greater than $\gamma_2 = 1$ are released in all zones. Thus, all zones remain with only the tasks in layers 0 and 1 fixed. In Fig. 4(c), following rule 3, the tasks that are released are those in layer 2 of zones 1 and 2, and those in layer 3 of zone 3, the tasks in the last layer ($\gamma_3 = 1$). As previously observed, $\gamma_2 = \gamma_3 = 1$ and the released tasks are different. This variant ends by calling (HM) to assign the released tasks to zones and to identify the three vehicle trips for each case ((a), (b), and (c)).

4.4. Overall procedure

To summarise the overall methodology we present a global flowchart in Fig. 5 and display the results for the example.

Example 5. The feasible solutions generated by all algorithms for Example 1 are compared in Table 2, showing different solutions even for this small example. The second column, HM_K, presents the values provided by the valid model (HM) with $\alpha = |K| - 1$, and in third to seventh columns are the values for the hybrid model matheuristic (H1); for the two-phase matheuristic (H2); and for its variant with rule (r) (H3(1), H3(2), H3(3)). From upper bound values, in the second row, we may see that the optimum was achieved by the valid model and the hybrid model matheuristic as well as by the 2-phase variant when fixing a percentage according to the demand (rule 1). Assuming all the tasks fixed, as in the 2-phase matheuristic (column 4), proved to be the worst option, in this small example. The third row compares the percentage of deadheading links in each solution, with the highest values corresponding to the worst solution values, as expected. The computation times (last row) are very small.

Table 2
Example results.

	HM_K	H1	H2	H3(1)	H3(2)	H3(3)
UB values	34[a]	34[a]	46	34[a]	38	40
deadheading links %	23.6	23.6	29.5	23.6	25.7	26.7
cpu time (s)	12.3	2.6	0.6	2.8	1.1	1.0

Legend:
[a]Optimal value.

5. Visual attractiveness and the weighted hull overlap

To qualify the visual attractiveness of trips (Lum et al., 2017) propose the hull overlap (HO) measure. Denoting by $FT = \{T_k\}$ the set of feasible trips and by C_k the convex hull of vertices visited by trip T_k, the overlapping of each trip, is defined as

$$HO_k = \sum_{k' \neq k} \frac{\text{Area}\left(C_k \cap C_{k'}\right)}{\text{Area}(C_k)} \tag{21}$$

and the global attractiveness of the solution is given by the average overlapping of the trips, that is,

$$HO = \frac{1}{|K|} \sum_{k=1}^{|K|} HO_k. \tag{22}$$

This measure combines information on overlapping with the placement of tasks relative to their ideal boundaries. In this work we

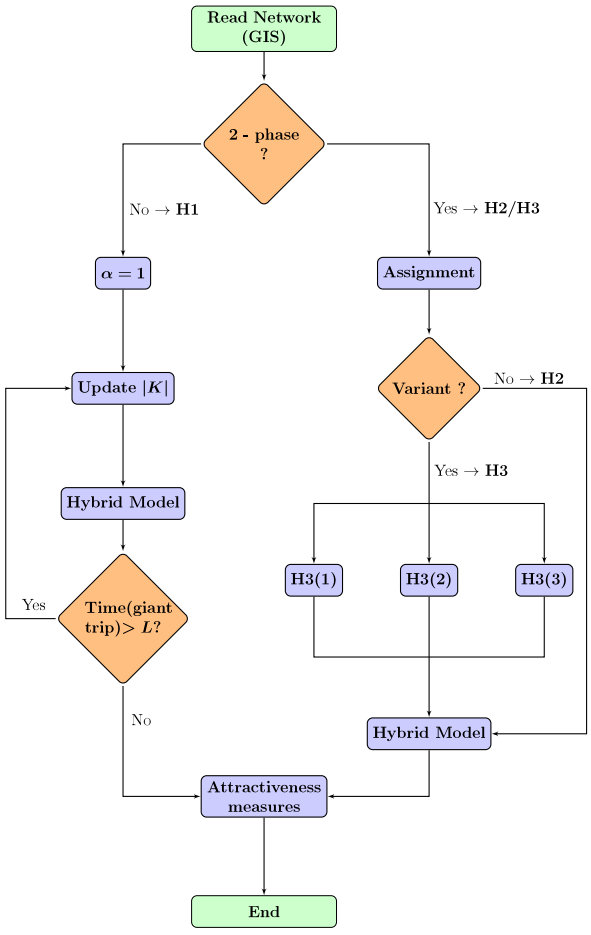


Fig. 5. Global illustration of the methodology.

considered that for an easier comparison between instances it would be more convenient to use normalised version of (21). Thus,

$$\overline{HO}_k = \frac{1}{|K|-1} \sum_{k' \neq k} \frac{\text{Area}\left(C_k \cap C_{k'}\right)}{\text{Area}(C_k)}, \tag{23}$$

and, consequently

$$\overline{HO} = \frac{1}{|K|} \sum_{k=1}^{|K|} \overline{HO}_k. \tag{24}$$

Thus $0 \leq \overline{HO} \leq 1$, where $\overline{HO} = 0$ means that there is no overlapping and $\overline{HO} = 1$ means that there is a full overlap of the convex hulls.

We propose a new attractiveness measure, the weighted hull overlap denoted by $\overline{HO}(\rho)$. This new measure is a generalisation of $\overline{HO}$, where we introduce space dependent penalisation of overlaps. It can be argued that $\overline{HO}$ over-penalises situations where hull overlaps occur in regions of low interest, such as, outside the road network area, while, at the same time, it under-penalises overlaps in sensitive regions. In order to address this issue, a penalty function, ρ, that weights the severity of the overlaps is introduced.

Let $\Omega \subset \mathbb{R}^2$ be a rectangle containing the network and consider an integrable function $\rho : \Omega \to \mathbb{R}$ such that $\int_\Omega \rho(x,y)dxdy = \text{Area}(\Omega)$. Adapting (23) by computing the area of the intersections weighted by ρ, we can define

$$\overline{HO}_k(\rho) = \frac{1}{|K|-1} \sum_{k' \neq k} \frac{\mu(C_k \cap C_{k'})}{\mu(C_k)}, \tag{25}$$

$$\overline{HO}(\rho) = \frac{1}{|K|} \sum_{k=1}^{|K|} \overline{HO}_k(\rho) \tag{26}$$

where

$$\mu(X) = \int_X \rho(x,y)dx\,dy, \quad X \subseteq \Omega.$$

This attractiveness indicator is an average overlap fraction, measured by μ. Generally speaking, ρ can be seen as a penalty function that evaluates the impact of an overlap depending on its location. One possible choice for ρ is to define it as the characteristic function of a certain region of interest, meaning that overlaps inside this region are valued with respect to overlaps occurring elsewhere. The density ρ can be specified either analytically or by directly assigning an increased weight to a set of nodes and extending it to Ω using low order piece-wise interpolation.

In general, the penalty function ρ should be further adjusted to incorporate information on sensitive areas where the overlapping is especially damaging. In Fig. 6 we represent three different choices of ρ for the same road network, and compute $\overline{HO}(\rho)$ in a small example with two trips.

- In Fig. 6(a), we use a constant penalty function ρ_1. The definition of $\overline{HO}(\rho)$ is equivalent to the one of $\overline{HO}$, thus $\overline{HO}(\rho_1) = \overline{HO} = 0.1$. This means that the area of overlap area is, on average, 10% of the convex hulls' area.
- In Fig. 6(b), we use a network density function, disregarding regions that are not close to the road network. The shadowed blue region, D, is composed of points within 250 m of road segments.

  $$\rho_2(x,y) = \begin{cases} cte, & (x,y) \in D \\ 0, & (x,y) \in \Omega \setminus D \end{cases}$$

  In this case, we are still computing areas, but excluding the parts of the convex hulls that fall out of D. We get $\overline{HO}(\rho_2) = 0.096$. This is quite similar to $\overline{HO}$ because the part of convex hulls that is outside D is small.
- In Fig. 6(c) we use a penalty function concentrated in two points, (x_1, y_1) and (x_2, y_2), of the form

  $$\rho_3(x,y) = c_1 + \sum_{i=1}^{2} c_i \exp\left(-\frac{1}{2}\frac{(x-x_i)^2}{\sigma_i^2} - \frac{1}{2}\frac{(y-y_i)^2}{\sigma_i^2}\right).$$

  In this case, since the overlap in the central point is highly penalised by ρ_3, we obtain $\overline{HO}(\rho_3) = 0.685$, which is much higher than the previous ones.

So, this specific trip design would be considered appropriate when using ρ_1 or ρ_2, but inadequate when using ρ_3. Generally speaking, the $\overline{HO}(\rho)$ measure carries more information than $\overline{HO}$ when there is good reason to penalise overlaps in specific parts of the road network.

Henceforward, the attractiveness of a solution is measured through $\overline{HO}$ and $\overline{HO}(\rho)$, in order to characterise its overlapping, as well as other measures (see Section 7.3), to qualify its connectivity and compactness. For the moment, the $\overline{HO}(\rho)$ index corresponds to a road network density function.

6. Geographic information system

Nowadays, GIS are widespread over several fields of activity, allowing the creation of personalised digital map layers that are essential for the solution of real-world problems. They are a key tool in regional and municipal planning, namely for land planning, provision of infrastructure, transportation networks design, mail distribution, and garbage collection, among others.

In this work we use the GIS of the municipality of Seixal, namely the layers containing road segments, traffic signs, and implanted buildings. The road segments are used to establish the connectivity and to design the road network, while the implanted buildings are assigned to each link and used to estimate the demand.

The road segment layer is a collection $(S_i)_{i=1}^n$, where each S_i stores the following information

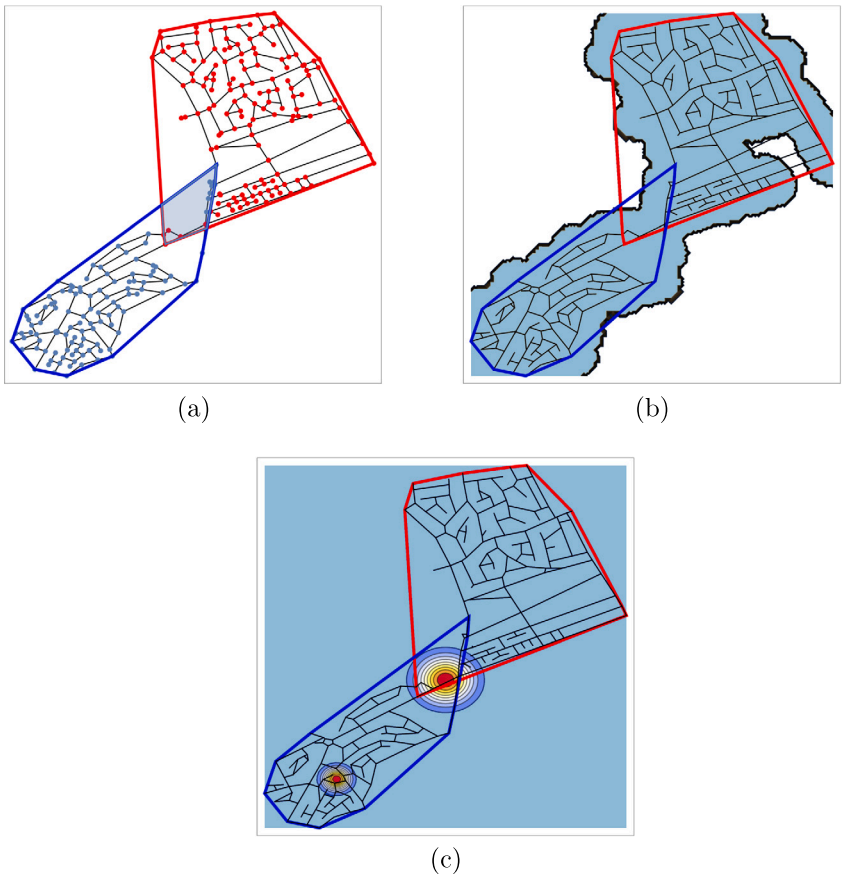


Fig. 6. (a) Two trips, represented by blue and red nodes and their convex hulls; (b) Network density function ρ_1; (c) Customised space dependent penalty ρ_2.

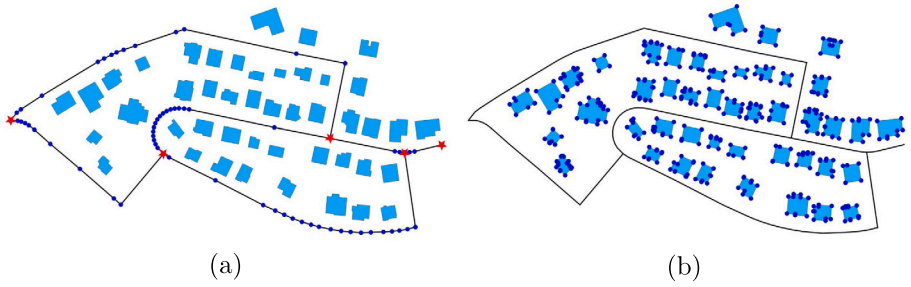


Fig. 7. (a) Road segment nodes (red stars) and auxiliary nodes (blue circles). (b) Geometrical representation of implanted buildings.

- `seg_eixo`: Unique identifier of the road segment [integer].
- `toponimo`: Unique identifier linking to the street name/code [string].
- `clas_rod`: Type of road (local paved road, local unpaved road, national road, highway, etc.) [string].
- `clas_actual`: Date of last update [string].
- `shape_len`: Length of segment [float].
- `shape_pts`: Array containing a sequence of geographical coordinates that define the road segment $(x_1, y_1), (x_2, y_2), \ldots, (x_{n_i}, y_{n_i})$ [array of float].

The connected graph is built by identifying common terminal points between different segments (see Fig. 7a). Each road segment will result in one arc (one-way street), two arcs (two-way street) or one edge (two-way street served with a zig-zag single pass).

The demand on each arc/edge is estimated from the number of buildings associated with that connection. Each building is assigned to a road segment, based on the proximity of its geometric centre to segment nodes (see Fig. 7b), and then to a street direction.

The municipality of Seixal covers 95 km^2, and has a population of over 160,000 inhabitants (see the Câmara Municipal do Seixal (CMS) web page, www.cm-seixal.pt/territorio) A door-to-door collection system was implemented in some districts, for both hygiene reasons and the population's comfort.

The full network is represented in Fig. 8, and it consists of 8909 road segments and 32,298 buildings, resulting in a graph with 6813 nodes, 11,838 arcs and 2991 edges. It is possible to graphically select subsets

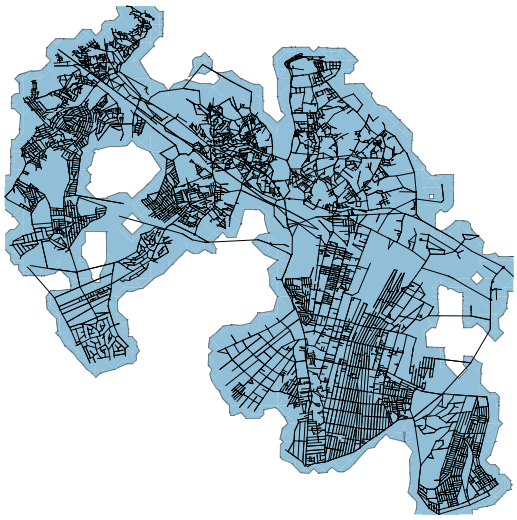


Fig. 8. The full Seixal network, using a grid resolution of 100.

Table 3
Characteristics of the Seixal instances.

Instance	$\|V\|$	$\|A' \cup E\|$	$\|A_R\|$	$\|E_R\|$	$\|K\|$	L	Q_T	CC_g
N1_2	265	492	119	138	2	25 200	29 940	11
N1_3					3	22 200		
N1_4					4	15 000		
N2_2	363	722	162	163	2	25 200	34 560	38
N2_3					3	22 200		
N2_4					4	15 000		
N3_3	467	933	207	216	3	25 200	45 030	45
N3_4					4	22 200		
N3_5					5	16 000		
N4_4	539	1119	259	272	4	35 000	83 715	40
N4_5					5	28 000		
N4_6					6	25 200		
N5_5	899	1783	429	458	5	41 000	104 925	56
N5_6					6	33 000		
N5_8					8	25 200		
N6_4	1223	2254	503	520	4	37 000	97 050	152
N6_5					5	33 000		
N6_6					6	25 200		

Legend: CC is the number of connected components in the demand graph.

of this network to yield smaller instances (see Table 3), motivated by preliminary information provided by the municipality. Once the trips are generated, they can be visualised back in the GIS interface.

7. Computational results

The performance of the proposed methodologies is analysed using lower and upper bound values, execution times, as well as from visual attractiveness measures, using the instances described next.

The integer programming models were solved with the IBM® ILOG® CPLEX® 12.9.0, in an AMD Ryzen Threadripper 2950X 16-Core Processor (64 GB RAM).

7.1. Problem instances

The instances were generated entirely and automatically using the GIS and accurately represent the current Seixal waste collection problem. This work follows that of Ghiani et al. (2014). Due to inconsistencies with today's reality, instances generated in this first preliminary project have been dropped. Although it used GIS to generate the network, namely regarding the refuse to collect, the network definition had an important manual part, e.g. the identification of the nodes. In the current work, nodes are geo-referenced by GIS. Moreover, since 2014, the waste system in Seixal has undergone several modifications, prompting for this new study, and practitioners point to the need to develop new instances.

The data set comprises six network instances, named N1_$|K|$–N6_$|K|$. Thus, by varying the number of vehicles, $|K|$ in the instance name, and the time limit per trip, a total of 18 instances were tried, as displayed in Table 3 (column 1). For each network a real time limit of 25,200 s (7 h) was essayed. Two other time limit values per instance were also considered to assess the performance of the proposed matheuristics. The network sizes vary, from 265–1223 nodes (including the depot and the landfill), and 492–2254 links (columns 2–3). The percentage of required arcs and edges is between 20% and 30% (columns 4 and 5). The number of vehicles, $|K|$, and the time limit considered per trip, L, are depicted in columns 6 and 7, respectively. The last two columns show the total demand, Q_T, and number of connected components, CC_g, of the graph induced by the set of required links.

7.2. Matheuristics results

As before, in this section, HM_K is used for results provided by the valid model (HM) with $\alpha = |K| - 1$, and matheuristics are represented by the acronyms: H1 designates the hybrid model matheuristic; H2 stands for the two-phase matheuristic; and H3(r) for its variant with rule r ($r = 1, 2, 3$). In the latter, parameter γ was set to γ_1, γ_2, or γ_3, when $r = 1, 2, 3$, respectively. Based on preliminary tests, γ_1 was set to 0.9 because, among the tested values (0.9, 0.8, 0.5, 0.3 and 0.1), it was the one providing better results: more feasible solutions (14 out of 18), five of them optimal, with an average gap from the instance lower bound equal to 10.02%. The results were also acceptable up to $\gamma_1 = 0.5$. Regarding the choice of γ_2, it is set in order to keep all the tasks in the zone with the smaller number of layers fixed, since, as before, it was the one leading to better results (12 feasible solutions, 5 optimal, and an average gap value of 9.48%). Finally, γ_3 is set to half of the difference between the maximum and the minimum number of layers in the instance. Again, this choice proved to be the best option (11 feasible solutions, 6 optimal ones and an average gap of 10.04%). Among values tested for γ_2 and γ_3 the chosen ones are those that release more layers and are still able to generate feasible solutions. Note that a smaller γ_2 (bigger γ_3) value leads to more released tasks, which increases the size of the MILP problem to be solved.

The hybrid model (HM) was used to obtain two lower bounds per instance. One, lb_K, results from the valid model. The other, lb_0, comes from the aggregated model that only considers the giant trip, that is $p = 0$ and no α. Thus, constraints (3), (7), (13), and (14), are eliminated. The valid model ran in CPLEX, within a pre-defined three hours time limit (TL).

Column two in Table 4 contains the gap between the two lower bound values, that is $gap_{lK\_l0} = (lb_K - lb_0)/lbbest \times 100$, with $lbbest = \max\{lb_K; lb_0\}$. Thus, a negative value points to a better performance of the aggregated model. As may be observed, all values but five are negative, with zero corresponding to the smallest instance where the valid model attains the optimal solution value, and the aggregated model proposes that same value for lower bound. The good performance of the aggregated model is even more impressive for the larger instances, with the corresponding lower bound exceeding up to 20% of lb_K.

Columns three to eight in Table 4 depict gap values, computed by

$$gap_{\bullet} = \frac{v(H\bullet) - lbbest}{lbbest} \times 100,$$

Table 4
Seixal results — gap values comparison.

Instance	gap_{lK_l0}	Gap values					
		HM_K	H1	H2	H3(1)	H3(2)	H3(3)
N1_2	0.00	0.00	0.00	2.96	2.50	2.96	2.96
N1_3	−0.12	0.52	0.41	5.99	5.77	5.99	5.99
N1_4	0.12	17.92	2.17	7.80	7.74	6.59	5.02
N2_2	1.14	4.57	4.57	5.23	3.92	4.95	5.23
N2_3	0.48	58.38	5.76	11.93	9.96	11.17	9.86
N2_4	1.04	–	7.52	12.26	10.39	11.74	22.14
N3_3	−0.44	–	–	12.65	11.36	10.79	12.23
N3_4	−0.65	121.95	6.99	15.67	14.36	15.15	13.59
N3_5	−0.82	–	12.74	18.91	18.63	–	–
N4_4	−9.83	–	–	21.39	18.32	20.20	19.13
N4_5	−6.97	–	11.43	25.10	–	–	–
N4_6	−7.53	–	16.91	27.26	–	–	–
N5_5	−4.17	246.74	0.27	8.99	7.75	8.15	8.19
N5_6	−6.62	–	0.17	8.97	8.46	7.86	5.93
N5_8	−6.36	–	0.29	8.68	7.45	8.09	–
N6_4	−14.79	–	–	12.20	–	–	–
N6_5	−20.34	–	–	14.01	13.48	–	–
N6_6	−17.65	–	–	16.49	–	–	–
#FS		7	13	18	14	12	11
#Best		1	12	2	3	1	0

Legend: #FS -no. of feasible solutions; #Best - no. of best solutions.

Table 5
Seixal results — execution times.

Seconds	Lower bounds		Upper bounds					
	lb_K	lb_0	HM_K	H1	H2	H3(1)	H3(2)	H3(3)
min	227.5	6.1	227.5	227.5	4.6	67.6	4.0	24.2
average	10 213.3	1518.8	9290.3	34,459.4	7341.3	7330.2	6854.6	6647.9
max	TL	TL	TL	64,805.3	TL	TL	TL	TL

Table 6
Seixal results — attractiveness measures, CI and ATD.

#instances		HM_K	H1	H2	H3(1)	H3(2)	H3(3)
		7	13	18	14	12	11
CI	min	8.7	10.0	5.0	5.0	4.5	5.3
	average	35.5	22.4	15.9	13.8	13.9	13.6
	max	63.2	43.0	41.3	35.0	22.0	22.0
ATD	min	106.7	81.6	40.2	40.5	56.2	55.8
	aver.	128.8	116.4	70.1	71.8	77.6	78.2
	max	147.0	161.3	121.1	122.0	128.4	212.1

where $v(H\bullet)$ is the upper bound achieved by matheuristic $H\bullet$ ($\bullet = M_K, 1, 2, 3(r)$). Remember that HM_K are the values provided by (HM) with $\alpha = |K| - 1$.

Note that the valid model proved to have found the optimal solution only once, and even for 11 out of 18 instances it did not encounter a feasible solution. However, we should stress that the aggregated model only identifies a giant trip so it does not provide feasible solutions for the overall problem, only lower bounds.

The (HM) with constraints (19) or (20), to balance the number of tasks or the served demand among zones, only achieved a few feasible solutions. Henceforward, only (HM) is analysed. However, for these few instances, it was noticed that zones provided by the model with the additional balance constraints are better balanced than the ones generated by Algorithm 3. This is due to the fact that some zones usually get stuck without neighbours in the first iterations while there are still a large number of free tasks, as the number of connected components is huge (see last column in Table 3). This also justified variant H3, where some tasks are released before the second phase.

As may be observed, the hybrid heuristic (H1) provides the best gap values more often, with the two-phase method (H2) always generating a feasible solution, within smaller computation times (see Table 5). Moreover, gap values for all two-phase matheuristics (H2 and H3(r)) are very similar, leading to the conclusion that releasing some tasks from the outer layers is not a good option. Note that the cpu times for H1 are quite large, as in each iteration it solves a MILP problem with a time limit of three hours. As expected, the valid model only achieves good solutions for small networks. Even when the three-hour time limit was increased, CPLEX did not succeed in finding feasible solutions.

Table 5 depicts the minimum, average and maximum cpu times (in seconds) regarding the model and the matheuristics. Thus, columns two and three present the times needed to reach valid and aggregated lower bounds, respectively. The last six columns show the times needed to attain the upper bounds for the correspondent matheuristic.

7.3. Attractiveness results

Attractiveness measures are useful to assess the suitability of solutions to be implemented in practice. Tables 6 and 7 present the results for the measures discussed in Section 5 as well as for the measures taken from the literature, which we shall now analyse.

The connectivity index (Constantino et al., 2015), $CI = CC/|K|$, is the average number of connected components (CC), and is always greater than or equal to one, and better the smaller it is. However, it would be unlikely for Seixal instances, with a number of connected components between 11 and 152, to present solutions with CI close to one. Thus, the values in Table 6 indicate that solutions may generally be considered as compact as possible, highlighting positively those of the two-phase matheuristics, particularly the variant H3(r). Recall that the first phase tries to join adjacent tasks in the same vehicle service (zone), which promotes smaller CI values. By releasing some tasks from their zones, H3(r) is able to further improve this index value.

The average task distance (Constantino et al., 2015), ATD, represents the average of the shortest deadheading path distance between tasks served by the same vehicle, and is given by:

$$ATD = \frac{1}{|K|}\sum_{k=1}^{|K|}\frac{1}{tpairs}\sum_{a,b \text{ in } k} D_{ab} \tag{27}$$

where $tpairs = |A_R \cup E_R| \times (|A_R \cup E_R| - |K|)/(2|K|^2)$ is used as an approximate value for the number of task pairs per trip, D_{ab} is the minimum deadheading distance from task a to task b, served in k, excluding d_a and d_b. From Table 6 it is clear that this measure also points to better solutions (smaller values) for the two-phase matheuristics, with similar values for all variants, and H2 performing slightly better. Thus, solutions generated by two-phase matheuristics are more compact.

The overlapping will be measured by $\overline{HO}$, $\overline{HO}(\rho)$, previously introduced, and by ROI (Constantino et al., 2015), which compares the node overlapping in the given solution with the one in an "ideal" solution. The attractiveness is higher the lower these measures are. ROI is computed as:

$$ROI = \frac{NO - |V|}{(\sqrt{|K|} + \sqrt{|V|} - 1)^2 - |V|} \tag{28}$$

where NO counts the number of nodes visited by the different trips and is given by $NO = \sum_{i \in V}\sum_{k=1}^{|K|} n_i^k$, being $n_i^k = 1$ iff node i is visited by vehicle k.

Table 7 presents these three measures for all the methodologies and instances. Similar to previous measures, the two-phase matheuristics display better values, keeping the H2 advantage, as H3 variants often present slight increases. It is interesting to note that the three measures often indicate the lowest values for the same methodology. Although ordering the methodologies equally, $\overline{HO}(\rho)$ is usually smaller than $\overline{HO}$ as it only penalises the areas of interest. Also worth noting is that for more general penalty functions $\overline{HO}(\rho)$ values can be greater than $\overline{HO}$. See, for instance the shadow area in Fig. 8, which represents the road network density function, equal to a constant value for points apart up

Table 7
Seixal results — overlapping measures.

inst.	HM_K			H1			H2			H3(1)			H3(2)			H3(3)		
	ROI	$\overline{HO}$	$\overline{HO}(\rho)$	ROI	$\overline{HO}$	$\overline{HO}(\rho)$	ROI	$\overline{HO}$	$\overline{HO}(\rho)$	ROI	$\overline{HO}$	$\overline{HO}(\rho)$	ROI	$\overline{HO}$	$\overline{HO}(\rho)$	ROI	$\overline{HO}$	$\overline{HO}(\rho)$
N1_2	1.06	0.57	0.26	1.06	0.57	0.26	0.23	**0.01**	**0.01**	**0.15**	**0.01**	**0.01**	0.23	0.04	0.02	0.23	0.04	0.02
N1_3	0.81	0.59	0.16	1.32	0.44	0.13	**0.30**	0.14	0.06	0.34	0.27	0.10	**0.30**	**0.13**	**0.05**	**0.30**	0.14	0.06
N1_4	2.13	0.70	0.22	1.51	0.53	0.13	0.43	**0.05**	**0.02**	0.40	**0.05**	**0.02**	**0.34**	0.12	0.06	0.65	0.32	0.14
N2_2	3.30	0.78	0.31	3.30	0.78	0.31	0.44	0.14	0.06	**0.32**	**0.12**	**0.04**	0.44	0.14	0.06	0.44	0.14	0.06
N2_3	4.27	0.71	0.34	1.03	0.35	0.12	0.28	0.13	0.05	**0.21**	0.14	0.06	0.46	0.16	0.07	**0.21**	**0.08**	**0.03**
N2_4	–	–	–	1.35	0.40	0.19	**0.44**	**0.13**	**0.06**	0.54	0.22	0.10	0.75	0.18	0.07	0.93	0.42	0.20
N3_3	–	–	–	–	–	–	0.22	**0.03**	**0.02**	0.26	0.19	0.05	0.45	0.22	0.06	**0.19**	**0.03**	**0.02**
N3_4	4.88	0.87	0.22	1.09	0.33	0.08	**0.23**	**0.06**	**0.02**	0.28	0.29	0.14	0.35	0.25	0.12	0.39	0.28	0.13
N3_5	–	–	–	1.51	0.39	0.09	**0.21**	**0.04**	**0.01**	0.24	0.08	0.03	–	–	–	–	–	–
N4_4	–	–	–	–	–	–	0.53	**0.06**	**0.02**	**0.45**	0.15	0.04	0.58	0.18	0.05	0.51	0.11	0.03
N4_5	–	–	–	2.13	0.46	0.15	**0.52**	**0.08**	**0.03**	–	–	–	–	–	–	–	–	–
N4_6	–	–	–	2.06	0.48	0.16	**0.54**	**0.06**	**0.02**	–	–	–	–	–	–	–	–	–
N5_5	4.43	0.82	0.26	1.99	0.69	0.19	0.54	0.07	0.03	0.50	0.07	**0.03**	0.53	**0.05**	**0.03**	**0.49**	0.06	0.03
N5_6	–	–	–	1.85	0.58	0.21	0.44	**0.09**	**0.05**	**0.43**	0.10	0.05	1.00	0.39	0.18	1.06	0.34	0.12
N5_8	–	–	–	1.86	0.55	0.21	**0.29**	**0.05**	**0.03**	0.37	0.12	0.06	0.86	0.50	0.21	–	–	–
N6_4	–	–	–	–	–	–	**0.20**	**0.06**	**0.02**	–	–	–	–	–	–	–	–	–
N6_5	–	–	–	–	–	–	0.29	**0.08**	**0.03**	**0.28**	0.12	0.05	–	–	–	–	–	–
N6_6	–	–	–	–	–	–	**0.28**	**0.06**	**0.02**	–	–	–	–	–	–	–	–	–
Min.	0.81	0.57	0.16	1.03	0.33	0.08	0.20	0.01	0.01	**0.15**	0.01	0.01	0.23	0.04	0.02	0.19	0.03	0.02
Average	2.98	0.72	0.25	1.70	0.50	0.17	0.36	**0.07**	**0.03**	**0.34**	0.14	0.06	0.52	0.20	0.08	0.49	0.18	0.08
Max.	4.88	0.87	0.34	3.30	0.78	0.31	**0.54**	**0.14**	**0.06**	**0.54**	0.29	0.14	1.00	0.50	0.21	1.06	0.42	0.20

to $d^* = 250$ m from the network, and zero otherwise. All tests were conducted using this road network density.

Matheuristic H1, generating better gap values, performs the worst regarding the attractiveness measures, as expected. The increase in average gap values from 5.3 for H1 to 13.1 for H2, corresponds to a decrease in average values from 22.4 to 15.9 for CI, from 116.4 to 70.1 for ATD, from 1.70 to 0.36 for ROI, from 0.50 to 0.07 for $\overline{HO}$, and from 0.17 to 0.03 for $\overline{HO}(\rho)$. Thus, the best option seems to be H2, which always finds a feasible solution, with better attractiveness measures, competitive cpu times, and gap values that do not deteriorate as much.

8. Conclusions

The household waste collection problem in the Portuguese municipality of Seixal was modelled as a generalisation of a mixed capacitated arc routing problem (MCARP). The proposed methodology to tackle the identified problem, included the use of GIS in the input/output phases and three matheuristics. All matheuristics include the resolution of a new hybrid model. The first one (H1) iteratively solves the developed new hybrid model. The remaining two, are versions of a two–phase matheuristic, that also uses GIS to assign tasks to layers in zones, one per vehicle, in the first phase. The hybrid model is then run in the second phase, assigning tasks differently in the two matheuristics (H2 and H3).

Instances for the Seixal waste collections problem are derived from six networks, with different dimensions. The quality of the generated solutions was assessed through both the total time (the objective value) and attractiveness measures, including a newly proposed one. The developed hybrid model simultaneously provided optimal solutions for small instances and good quality lower bounds.

Matheuristic H1, usually found better gap values sacrificing the values of attractiveness measures, as expected. The increase in average gap values from H1 to H2, corresponded to a decrease in average values for all attractiveness measures. Moreover, H1 failed to find feasible solutions for larger instances, which was not the case for H2, that was also able to produce results for all instances in smaller computation times.

The final option, among the two-phase matheuristics, points again to H2. The results generated from H2 and H3(r) (r = 1, 2, 3) may be considered very similar for both gap values and attractiveness measures, within identical computation times. H3(r), using different rules to release the assignment of some tasks for the second phase, also failed to find feasible solutions for the larger instances.

The generated trips are still to be tested in practice, which will be hopefully done in a restricted zone in the near future.

This work indicates the advantage of using a hybrid model embedded in a two-phase matheuristic. Future research could explore how to draw on GIS more often during the methodology, in order to not only better repair attractiveness values but also to better explore the hybrid model's capabilities. Additionally, some future work relating the attractiveness measures work could also be pursued.

CRediT authorship contribution statement

João Janela: The original idea, Methodology, Modelling choices, Design of computational solutions. **Maria Cândida Mourão:** The original idea, Methodology, Modelling choices, Design of computational solutions. **Leonor Santiago Pinto:** The original idea, Methodology, Modelling choices, Design of computational solutions.

Funding

This work was partially supported by the Projects CEMAPRE/REM - UIDB /05069/2020 and CMAFcIO - UIDB/04561/2020, financed by FCT/MCTES through national funds.

References

Archetti, C., Speranza, M., 2014. A survey on matheuristics for routing problems. EURO J. Comput. Optim. 2, 223–246.

Bodin, L., Fagin, G., Welebny, R., Greenberg, J., 1989. The design of a computarized sanitation vehicle routing and scheduling system for the town of Oyster Bay, New York. Comput. Oper. Res. 16, 45–54.

Braier, G., Durán, G., Marenco, J., Wesner, F., 2017. An integer programming approach to a real-world recyclable waste collection problem in Argentina. Waste Manage. Res. 35, 525–533.

Chu, F., Labadi, N., Prins, C., 2005. Heuristics for the periodic capacitated arc routing problem. J. Intell. Manuf. 16, 243–251.

Constantino, M., Gouveia, L., Mourão, M., Nunes, A., 2015. The mixed capacitated arc routing problems with non-overlapping routes. European J. Oper. Res. 244, 445–456.

Corberán, A., Eglese, R., Hasle, G., Plana, I., Sanchis, J., 2021. Arc routing problems: A review of the past, present, and future. Networks 77, 88–115.

Corberán, A., Golden, B., Lum, O., Plana, I., Sanchis, J., 2017. Aesthetic considerations for the min-max K-windy rural postman problem. Networks 70, 216–232.
Corberán, Á., Laporte, G. (Eds.), 2014. Arc Routing: Problems, Methods and Applications. In: MOS-SIAM Series on Optimization, Boston Philadelphia.
Corberán, A., Prins, C., 2010. Recent results on arc routing problems: An annotated bibliography. Networks 56, 50–69.
Dror, M., 2000. Arc Routing: Theory, Solutions and Applications. Kluwer Academic, Boston.
Ghiani, G., Mourão, M., Pinto, L., Vigo, D., 2014. Routing in waste collection applications. In: Corberán, A., Laporte, G. (Eds.), Arc Routing: Problems, Methods and Applications. In: MOS-SIAM Series on Optimization, pp. 351–370, (Ch. 15).
Golden, B., Assad, A., Wasil, E., 2002. Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM, Philadelphia, pp. 245–286.
Golden, B., Wong, R., 1981. Capacitated arc routing problems. Networks 11, 305–315.
Gouveia, L., Mourão, M., Pinto, L., 2010. Lower bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. 37, 692–699.
Kiilerich, L., Wøhlk, S., 2018. New large-scale data instances for CARP and new variations of CARP. INFOR: Inform. Syst. Oper. Res. 56, 1–32.
Lum, O., Cerrone, C., Golden, B., Wasil, E., 2017. Partitioning a street network into compact, balanced, and visually appealing routes. Networks 69 (3), 290–303.
Malakahmad, A., Bakri, P., Mokhtar, M., Khalil, N., 2014. Solid waste collection routes optimization via GIS techniques in Ipoh city, Malaysia. Procedia Eng. 77, 20–27.
Martins, K., Mourão, M., Pinto, L., 2013. A routing and waste collection case-study. In: Almeida, J., Oliveira, J., Pinto, A. (Eds.), Operational Research. Springer, pp. 261–276.
Monroy, I., Amaya, C., Langevin, A., 2013. The periodic capacitated arc routing problem with irregular services. Discrete Appl. Math. 161, 691–701.
Mourão, M., Pinto, L., 2017. An updated annotated bibliography on arc routing problems. Networks 70 (3), 144–194.
Santos, L., Coutinho-Rodrigues, J., Current, J., 2008. Implementing a multi-vehicle multi-route spatial decision support system for efficient trash collection in Portugal. Transp. Res. A 42, 922–934.
Shuster, K., Schur, D., 1974. Heuristic routing for solid waste collection vehicles. Federal Solid Waste Manage. Prog. SW-113, 1–45.
Sniezek, J., Bodin, L., 2006. Using mixed integer programming for solving the capacitated arc routing problem with vehicle/site dependencies with an application to the routing of residential sanitation collection vehicles. Ann. Oper. Res. 144, 33–58.
Sniezek, J., Bodin, L., Levy, L., Ball, M., 2002. The capacitated arc routing problem with vehicle/site dependencies: the philadelphia experience. In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics an its Applications, Philadelphia, pp. 287–308.
Tirkolaee, E., Hosseinabadi, A., Soltani, M., Sangaiah, A., Wang, J., 2018. A hybrid genetic algorithm for multi-trip green capacitated arc routing problem in the scope of urban services. Sustainability 10, 1366–1387.
Zbib, H., Laporte, G., 2020. The commodity-split multi-compartment capacitated arc routing problem. Comput. Oper. Res. 122, (published online).