Annals of Operations Research (2025) 350:979–1015
https://doi.org/10.1007/s10479-025-06626-4

ORIGINAL RESEARCH

Modeling and solving an integrated periodic vehicle routing and capacitated facility location problem in the context of solid waste collection

Begoña González[1] · Diego Rossit[2] · Mariano Frutos[3] · Máximo Méndez[1]

Received: 10 June 2024 / Accepted: 14 April 2025 / Published online: 18 May 2025

Abstract

Few activities are as crucial in urban environments as waste management. Mismanagement of waste can cause significant economic, social, and environmental damage. However, waste management is often a complex system to manage and therefore where computational decision-support tools can play a pivotal role in assisting managers to make faster and better decisions. In this sense, this article proposes, on the one hand, a unified optimization model to address two common waste management system optimization problem: the determination of the capacity of waste bins in the collection network and the design and scheduling of collection routes. The integration of these two problems is not usual in the literature since each of them separately is already a major computational challenge. Two improved exact formulations based on mathematical programming and two metaheuristic methods are provided to solve this proposed unified optimization model. It should be noted that the metaheuristics consider a mixed chromosome representation of the solutions combining binary and integer alleles, in order to solve realistic instances of this complex problem. Different parameters of the metaheuristics considered – a Genetic Algorithm and a Simulated Annealing algorithm – have been tested to study which combination of them obtained better results in execution times on the order of that of the exact solvers. The achieved results show that the proposed metaheuristic methods perform efficient on large instances, where exact formulations are not applicable, and offer feasible, high-quality solutions in reasonable calculation times.

✉ Begoña González
bego.landin@ulpgc.es

✉ Diego Rossit
diego.rossit@uns.edu.ar

Mariano Frutos
mfrutos@uns.edu.ar

Máximo Méndez
maximo.mendez@ulpgc.es

1 Instituto Universitario SIANI, Universidad de Las Palmas de Gran Canaria, Parque Científico y Tecnológico, Campus Universitario de Tafira, Las Palmas de Gran Canaria 35017, Canarias, Spain

2 INMABB, Departament of Engineering, Universidad Nacional del Sur-CONICET, 1253 Alem Av., Bahía Blanca 8000, Argentina

3 IIESS, Departament of Engineering, Universidad Nacional del Sur-CONICET, 1253 Alem Av., Bahía Blanca 8000, Argentina

Keywords Waste management · Periodic capacitated vehicle routing problem · Capacitated facility location problem · Mixed integer programming · Genetic algorithms · Simulated annealing algorithms

1 Introduction

Today, cities face numerous challenges to ensure the livability of the urban environment and the well-being of their citizens. One critical aspect that demands accurate handling is the proper collection of waste generated as part of the various city activities. A primary source of this waste is households, which contribute to what is known as Municipal Solid Waste (MSW). MSW is known for its heterogeneity, as it comprises different types of materials and is characterized by a distributed generation (Rossit et al., 2017). In contrast to other types of waste, such as industrial or sanitary waste, which are typically produced in large quantities at specific locations, MSW is generated at numerous city locations in smaller amounts. This poses a logistical challenge for authorities tasked with collecting this waste, often on a daily or two-day basis, from several scattered sources.

When it comes to collecting waste from households, various systems can be implemented. One traditional approach is the door-to-door system, where waste is collected individually from each household. This method, still prevalent in many cities, involves the collection vehicle visiting every household. Another common method is the community bin system, in which citizens deposit waste in bins distributed throughout the city. The points where these community bins are located, commonly referred to as *collection points*, are visited, not necessarily every day, by collection vehicles that empty the bins. Recent research suggests that the use of a point-based system is preferable to door-to-door collection in crowded places, as it can enhance the cost-efficiency of the logistics of the system, for example, by minimizing the distances that waste collection trucks have to travel (Rossi et al., 2022; Rossit & Nesmachnow, 2022). This logistic cost reduction may be especially notable in nations with already elevated logistics costs, such as Argentina (Asociación Latinoamericana de Logística, 2023), and in particular for the city of Bahía Blanca, which exceeds the national average in waste management expenditure (Rossit et al., 2023), and from which the case study presented in this paper were taken.

The establishment of the collection point network in the community bin system involves determining the capacity of these locations. Thus, the decision regarding the capacity of collection points is closely interconnected with the facet of the MSW system that comes immediately after in the reverse logistics chain, that is, the waste collection. Management of MSW collection includes the planning of routes, that is, the actual paths traveled within city streets to collect the waste deposited at these collection points, and the arrangement of these routes within a planning timeframe to create a regular, usually weekly, schedule. These two problems: the determination of collection point capacity and the design of collection routes, are interrelated. Thus, given a waste generation rate, the storage capacity of the collection points will dictate the frequency of collection truck visits to prevent overflow (Mahéo et al., 2020). However, both problems are usually solved separately due to the computational complexity that an integrated approach would imply. In fact, only solving the point-based network design involves solving a capacitated facility location problem, which is known to be an NP-hard problem (Cornuéjols et al., 1991). In developing countries, these issues are often addressed based on practitioner experience rather than computer-aided tools (Maalouf & Agamuthu, 2023), including Argentina (Cavallin et al., 2020). While practical experience is valuable, the academic literature offers insights into cost minimization and environmental preservation through computer-aided decision support methodologies.

Thus, the aim of this study is to propose an integrated approach to simultaneously address the design of both network and collection routes for MSW, involving periodic scheduling of routing, which is an area where there is still a vacancy in the related literature constituting a research gap. In this line of research, this paper proposes a novel mathematical formulation of the problem, as well as a linear and a quadratic mathematical programming model, and a mixed representation of the solutions to solve the target problem with metaheuristic methods. Computational experimentation with real-world instances is also provided to validate the suitability of this mixed representation of the solutions. Specifically, two evolutionary computation methods were chosen: the one most often used to solve the vehicle routing problem (VRP) and its variants (Elshaer & Awad, 2020): Genetic Algorithm (GA), and one rarely used: Simulated Annealing (SA), which has proven to be very competent in solving the problem addressed in this paper.

The remainder of the paper is organized as follows: Section 2 provides a review of the related literature and outlines the main contributions of this work to the state-of-the-art in the subject. Section 3 presents both the conceptual description and the mathematical formulation of the problem addressed. Section 4 describes the resolution approaches considered to solve the proposed mathematical model. Section 5 presents the computational experimentation, including the description of instances and the parametrization of the metaheuristic methods considered. Section 6 shows the numerical results achieved with both resolution approaches and performs a comparative analysis of them. Finally, Section 7 presents the conclusions and future work.

2 Literature review

As mentioned above, the integrated problem of simultaneously determining the capacity of collection points and designing collection routes has received limited attention in the literature. Hemmelmayr et al. (2014) proposed the first model based on mathematical programming that simultaneously determines the capacity of waste bins alongside waste collection routes and their scheduling. They proposed different conceptual models of the bin location part considering whether there is an initial configuration of bins to be relocated or not, considering whether there are space limits, and considering whether there is a waste source classification. They solved small synthetic instances with an exact solver and complemented it with a Large Neighborhood Search (LNS) algorithm for solving real-world instances of an unspecified city in Italy.

In the context of recyclable waste, Hemmelmayr et al. (2017) presented a location-routing approach that aims to simultaneously optimize the location of collection points, the capacity of bins to be installed within each collection point and the routing schedule. They proposed a mathematical formulation for solving small instances and an adaptive LNS for larger instances, which is able to obtain near-optimal solutions in small instances. The largest instance they addressed considered a maximum of 50 collection points. Also in the context of recyclable waste, Cubillos and Wøhlk (2020) proposed a similar model that was addressed with a hybrid Variable Neighborhood Search (VNS) algorithm. In this case, it should be noted that the waste collection problem was limited to a single day, so it did not involve periodic routing, and that the authors chose an uncapacitated vehicle for waste collection, so it could collect all the garbage in a single trip. All of this is a major simplification that reduces the complexity of the problem.

Subsequently, Gläser and Stücken (2021) proposed a model that extended the problem of Hemmelmayr et al. (2014) by introducing predefined alternatives for bin collection frequencies and considering user allocation to collection points. They presented an adaptive LNS heuristic to solve the problem, which could outperform the results of plain CPLEX.

In a related context, Roy et al. (2022) presented a model that tackles waste bin allocation and vehicle routing using VNS with Ant Colony Optimization, incorporating smart bins with alert capability for imminent filling. They analyze the presented model with realistic instances of Seoul (Korea) with 15 collection points, concluding that the model is effective in finding good quality solutions in a reasonable time, however, it should be noted that their model does not incorporate decision making on what type and capacity of bins to place at each collection point.

Lastly, Mahéo et al. (2023) presented integrated models to solve a similar allocation-routing problem, employing a novel resolution method based on Benders decomposition and valid cuts. The proposed method is able to outperform plain CPLEX in realistic instances, with at most 7 collection points. As in previous works, the authors considered predefined alternatives for bin collection frequencies.

Recently, other studies have addressed similar location-routing problems within the reverse logistic chain of municipal waste. For instance, Han et al. (2024) focused on a location-routing problem in which waste generation is grouped by residential sectors. Their model integrates decisions such as transporting waste from residential sectors to transfer stations using small vehicles, and then collecting waste at transfer stations and transporting it to disposal centers using large vehicles. However, their proposed model, which aims to minimize the distance traveled by both types of vehicles, does not take into account either the capacity restrictions of the facilities or a periodic schedule for routing. To solve this problem, the authors presented a mathematical formulation and used an approach based on the Dantzig-Wolfe decomposition, which demonstrated superior performance compared to plain CPLEX in real-world instances of Nanchang, China. For a similar problem but considering a stochastic waste generation rate following a known normal distribution, Niu et al. (2024) presented a multi-objective model to simultaneously optimize cost, greenhouse gas emission and citizens' satisfaction. However, neither the capacity of the facilities nor a periodic schedule of the routing were considered. To solve the optimization problem, they proposed an evolutionary algorithm hybridized with a decision tree, which is able to outperform other classical multi-objective evolutionary algorithms in real-world instances of the Chinese city of Beijing. The main characteristics of the previous related articles are summarized in Table 1.

2.1 Contribution of this work

After a thorough review of the literature related to MSW collection by the community bin system, it has been observed that the integrated problem of simultaneously determining the capacity of collection points and designing collection routes has received limited attention. In this framework, the present document contributes to the state-of-the-art with the following items:

1. *Expansion of the conceptual model*. The conceptual model considered in other works is extended by including decisions on the capacity of the facilities and avoiding predefined alternatives for MSW collection frequencies. This approach, although it implies an increase in computational complexity, allows greater flexibility both in choosing the

Table 1 Previous related articles

Author(s)	Resolution method	Case study	Main features and limitations
Cubillos and Wøhlk (2020)	Variable Neighborhood Search (VNS) algorithm	Five districts of Denmark	• collection is solved for a single-day, • no capacity restrictions for vehicles are contemplated.
Gläser and Stücken (2021)	LNS heuristic	-	• allocation of generators to collection points is considered, • the collection schedule is fixed in advance.
Han et al. (2024)	Branch-and-price, column generation	Nanchang, China	• allocation of generators to collection points is considered, • collection points capacity or periodic schedule for routing is not contemplated.
Hemmelmayr et al. (2014)	Mathematical model and metaheuristic	Italy	• the collection schedule is fixed in advance.
Hemmelmayr et al. (2017)	Adaptive Large Neighborhood Search	-	• instances up to 50 collection points, • it is specific for recyclable waste.
Mahéo et al. (2020)	Benders decomposition	Bahía Blanca, Argentina	• the collection schedule is fixed in advance.
Mahéo et al. (2023)	Benders decomposition and valid cuts	Bahía Blanca, Argentina	• the collection schedule is fixed in advance.
Niu et al. (2024)	Decision tree classifier	Beijing, China	• multi-objective approach considering cost, greenhouse gas emission and citizens' satisfaction, • capacity of the facilities nor a periodic schedule of the routing were considered.
Roy et al. (2022)	Hybrid VNS- ACO	Seoul, Korea	• smart bins are considered, • instances up to 15 collection points, • capacity of collection points is not contemplated.

bin combination to be placed at each collection point and in establishing the collection frequency of MSW accumulated in each of them.

2. *Novel mathematical formulation*. A mathematical formulation with new features is proposed to formalize the extended problem that began to be developed in Rossit et al. (2024). Thus, unlike other works (see Table 1) the proposed mathematical formulation incorporates the periodic nature of the problem, optimizing non-homogeneous waste collection schedules covering several days, generating a flexible scheduling of waste collection services at collection points, where waste is allowed to accumulate over several days, as the capacity of the accumulation points is simultaneously optimized. It also incorporates realistic capacity constraints for both collection points and vehicles and a fleet size is set.

Furthermore, a Mixed Integer Quadratic Programming (MIQP) model and a linearized Mixed Integer Linear Programming (MILP) model were developed and used with exact solvers to address the problem.

3. *Proposal of a mixed representation of the solutions to solve the target problem with metaheuristic methods*. Since it is widely recognized that metaheuristic approaches are suitable for solving computationally complex optimization problems, a Genetic Algorithm and a Simulated Annealing algorithm, both with a mixed binary-permutation encoding, are proposed to solve the target problem.
4. *Computational experimentation with real-world instances*. We start by comparing the performance of the two resolution approaches considered (exact solvers versus metaheuristics) using relatively small instances, in line with other previous studies using exact approaches, since this type of approach is usually only applicable for not very large instances. We end with an evaluation of the ability of metaheuristics to solve real-world instances using one with 163 collection points corresponding to an area of the Argentinean city of Bahía Blanca. It should be noted that the size of this instance is considerable compared to the instances considered in other works reviewed in the related literature.

3 The integrated periodic vehicle routing and capacitated facility location problem in the context of MSW collection

The optimization problem addressed in this section arises by simultaneously considering two different aspects of the MSW reverse logistics chain, namely:

1. the waste bin allocation problem to set the types and number of bins to be installed in predefined locations, according to the local waste generation, and the visit schedule of the collection vehicle.
2. the waste collection problem to define the collection points to be visited each day, the visit order of these points and the scheduling of the daily routes in the planning horizon (since not all collection points must to be visited every day). This corresponds to a Periodic Capacitated Vehicle Routing Problem (PCVRP) (Beltrami & Bodin, 1974).

This joint approach represents an inventory-routing problem (Archetti and Ljubić, 2022) with the additional feature that the storage capacity of each collection point is not predefined, instead it must be determined within the optimization process to minimize both the cost of the routing schedule and the cost of bin installation and maintenance. Additionally, realistic constraints are taken into account both for bin allocation and for the design of the daily routes in the planning horizon considered. These constraints include: a discrete number of capacity alternatives for bin combinations at collection points derived from the type of commercial bins available, a limit on the waste accumulation at collection points to avoid exceeding the capacity of the bins installed, a restriction on the amount of waste collected by each collection vehicle on each route, to ensure that it does not exceed its capacity, a limitation on the number of vehicles required per day that not exceed the fleet size, and the guarantee that the time required to complete each route from the departure of the collection vehicle from the depot to its return, including service times at each collection point and unloading time at the depot, does not exceed the drivers' work shift.

The mathematical model proposed extends the work of Mahéo et al. (2020, 2023) in that it does not simply consider a predefined set of feasible visit combinations to collect waste from collection points, but rather any possible visit combination can be used as long as the collection point sites do not overflow (i.e., as long as the waste accumulated at each collection point does not exceed the capacity of the bin combination chosen for that point). This is a feature that differentiates it from other models proposed in relevant papers addressing similar problems in the literature (Gläser and Stücken, 2021; Hemmelmayr et al., 2014). In addition, the time of service at each collection point is not fixed as in Mahéo et al. (2020, 2023), but depends on the type and number of bins installed.

Let n_I, n_B, n_V and n_T be the number of collection points, bin combinations, collection vehicles and days in the planning horizon, respectively. A collection point, $i \in I = \{1, 2, ..., n_I\}$, is a predefined location in an urban area where waste bins can be installed. We define $I^0 = I \cup \{0\}$, where 0 represents the depot where the collected waste is deposited and where each vehicle, $v \in V = \{1, 2, ..., n_V\}$, starts and ends its daily routes. A bin combination, $b \in B = \{1, 2, ..., n_B\}$, is a set of bins that can be installed at a collection point, taking into account physical space constraints. Therefore, each element of B can consist of a single bin or a combination of bins. The method used to determine the bin combinations considered in this paper can be found in Mahéo et al. (2023). Lastly, subset $T' \subset T = \{1, 2, ..., n_T\}$ is defined as the days of the planning horizon on which waste collection is not carried out due to drivers' rest days, for example, Sundays in the city of Bahía Blanca, Argentina

The following parameters are also defined:

Q: Collection vehicle capacity (m^3).
C_{ij}: Travel time (minutes) between points $i \in I^0$ and $j \in I^0$.
S_b: Service time (minutes) of bin combination $b \in B$.
T_U: Unloading time (minutes) of collection vehicles at the depot.
W_i: Amount of waste deposited daily (m^3/day) at collection point $i \in I$.
CAP_b: Capacity (m^3) of bin combination $b \in B$.
CIN_b: Adjusted cost over the planning horizon (US$) of bin combination $b \in B$, taking into account both acquisition and installation costs, as both are to be amortized over the useful life of the bins.
C_{CV}: Cost of collection vehicles (US$/min).
T_L: Length of working day for drivers (minutes).

Waste collection is assumed to take place at the end of the day, once all waste generated (daily) has been deposited at the corresponding collection points. With this in mind, the following variables are also defined:

x_{ijvt}: 1 if vehicle $v \in V$ travels from point $i \in I^0$ to point $j \in I^0$ on day $t \in T$, and 0 otherwise.
y_{ijvt}: load of vehicle $v \in V$ when it travels from point $i \in I^0$ to point $j \in I^0$ on day $t \in T$.
w_{it}: amount of waste accumulated at collection point $i \in I$ at the end of day $t \in T$.
w_i^{max}: maximum daily amount of waste accumulated at collection point $i \in I$ in the planning horizon.
n_{bi}: 1 if bin combination $b \in B$ is selected for collection point $i \in I$, and 0 otherwise.

Finally, Eq. (1) calculates TT_{vt}, the total service time of vehicle $v \in V$, on day $t \in T$, from the time it leaves the depot until it returns to it, including the service times at each collection point visited and the unloading time at the depot.

$$TT_{vt} = T_U \sum_{i \in I} x_{i0vt} + \sum_{i \in I^0} \sum_{j \in I^0} x_{ijvt} \left(C_{ij} + \sum_{b \in B} S_b n_{bj} \right) \tag{1}$$

Taking all these elements into account, the following mathematical model is proposed:

Minimize

$$\sum_{b\in B}\left(CIN_b\sum_{i\in I}n_{bi}\right)+C_{CV}\sum_{t\in T}\sum_{v\in V}TT_{vt} \tag{2}$$

subject to

$$n_{b0}=0,\ \forall b\in B \tag{2a}$$

$$\sum_{b\in B}n_{bi}=1,\ \forall i\in I \tag{2b}$$

$$\sum_{b\in B}CAP_b n_{bi}\geq w_i^{max},\ \forall i\in I \tag{2c}$$

$$x_{iivt=0}\ \forall i\in I^0, v\in V, t\in T \tag{2d}$$

$$x_{ijvt}=0\ \forall i,j\in I^0, v\in V, t\in T' \tag{2e}$$

$$\sum_{i\in I^0}x_{ijvt}-\sum_{i\in I^0}x_{jivt}=0,\ \ j\in I^0, v\in V, t\in T \tag{2f}$$

$$\sum_{i\in I}x_{0ivt}\leq 1,\ \ \forall v\in V, t\in (T-T') \tag{2g}$$

$$TT_{vt}\leq T_L,\ \ \forall v\in V, t\in (T-T') \tag{2h}$$

$$y_{ijvt}\leq Qx_{ijvt},\ \ \forall i,j\in I^0, v\in V, t\in T \tag{2i}$$

$$\sum_{i\in I^0}y_{ijvt}+w_{jt}\leq\sum_{i\in I^0}y_{jivt}+Q\left(1-\sum_{i\in I^0}x_{ijvt}\right), \quad \forall j\in I, v\in V, t\in T \tag{2j}$$

$$w_{it}=W_i+w_{i(t-1)}\left(1-\sum_{j\in I^0}\sum_{v\in V}x_{ijv(t-1)}\right), \quad \forall i\in I, t\in (T-\{1\}) \tag{2k}$$

$$w_{i1}=W_i+w_{in_T}\left(1-\sum_{j\in I^0}\sum_{v\in V}x_{ijvn_T}\right),\ \forall i\in I \tag{2l}$$

$$w_{it}\leq w_i^{max},\ \forall i\in I, t\in T \tag{2m}$$

where (in matrix notation)

$$\begin{aligned}
&x=[x_{ijvt}]\in\mathcal{M}_{(n_I+1)\times(n_I+1)\times n_V\times n_T}(\{0,1\}),\\
&y=[y_{ijvt}]\in\mathcal{M}_{(n_I+1)\times(n_I+1)\times n_V\times n_T}\left(\mathbb{R}^+\cup\{0\}\right),\\
&w=[w_{it}]\in\mathcal{M}_{n_I\times n_T}\left(\mathbb{R}^+\cup\{0\}\right),\\
&w^{max}=[w_i^{max}]\in\mathcal{M}_{n_I}\left(\mathbb{R}^+\cup\{0\}\right),\\
&n=[n_{bi}]\in\mathcal{M}_{n_B\times(n_I+1)}(\{0,1\}).
\end{aligned}$$

The mathematical model provided simultaneously addresses the problems of MSW collection and bin combination selection. Thus, the objective function (2) calculates the overall cost of installation and maintenance of bins and MSW collection. The latter based on the total service time of the vehicles (1). Regarding the restrictions, Eq. (2a) ensures that the model does not select a bin combination for the depot while Eq. (2b) states that only one bin combination can be placed at each collection point. Eq. (2c) establishes that the storage capacity installed at each collection point has to be sufficient to contain the maximum quantity of waste accumulated at that point over the planning horizon. Eq. (2d) shows that vehicles always move between different collection points and, once they have left the depot, they must visit at least one collection point before returning to the depot. Eq. (2e) shows that vehicles are out of service on drivers' rest days. Eq. (2f) states that each vehicle has to leave each visited collection point. Eq. (2g) establishes that each vehicle may perform one or no routes per day. Therefore, the number of daily routes cannot exceed the size of the vehicle fleet. Eq. (2h) prevents that each route, from the time a collection vehicle leaves the depot until it returns to the depot, including service times at each collection point visited and unloading time at the depot, can be performed in the working day. Eq. (2i) forces not to exceed vehicle capacity on collection routes. Eq. (2j) is a subroute elimination restriction that also keeps track of vehicle loading by updating variable y_{ijvt} (Toth & Vigo, 2014). Eqs. (2k) and (2l) keep track of the waste accumulated at each collection point at the end of each day, which depends on the waste generated on the day and whether any collection vehicle has visited that point on the previous day. For example, in Eq. (2k), if a collection point i is not visited the previous day, $t-1$, then $\sum_{j \in I^0} \sum_{v \in V} x_{ijv(t-1)} = 0$ and Eq. (2k) is reduced to $w_{it} = W_i + w_{i(t-1)}$. Conversely, when a collection point i is visited the previous day, $t-1$, then $\sum_{j \in I^0} \sum_{v \in V} x_{ijv(t-1)} = 1$ and Eq. (2k) is reduced to $w_{it} = W_i$. Specifically, Eq. (2l) defines the cyclic characteristic of planning since, on the first day of the planning horizon, the residuals accumulated on the last day of the planning horizon has to be considered, in order for the solution to be feasible when the problem is implemented as a PCVRP. Eq. (2m) sets, for each collection point, the maximum amount of waste accumulated each day of the planning horizon, in order to estimate the capacity of the bin combination to be installed at each collection point (linked to Eq. (2c)).

4 Resolution approaches

Two different approaches have been chosen to solve the problem described in Section 3. Specifically, two exact models and two metaheuristic methods have been considered.

4.1 Exact methods

The mathematical model described in Section 3 is a MIQP formulation due to the presence of a bilinear term both in the expression to compute the travel time TT_{vt} (1) which affects the objective function (2) and the time limit restriction (2h), and in restrictions (2k) and (2l). However, the latter two can be replaced by Eqs. (3a)–(3c) which no longer have bilinear terms. This is a different form of linearization of Eqs. (2k) and (2l) than the one proposed by Rossit et al. (2024), which involved the inclusion of additional variables.

$$w_{it} \geq W_i + w_{i(t-1)} - BigM \sum_{j \in I^0} \sum_{v \in V} x_{ijv(t-1)}, \ \forall i \in I, t \in (T - \{1\}) \tag{3a}$$

$$w_{i1} \geq W_i + w_{in_T} - BigM \sum_{j \in I^0} \sum_{v \in V} x_{ijvn_T}, \ \forall i \in I \tag{3b}$$

$$w_{it} \geq W_i, \ \forall i \in I, \forall t \in T \tag{3c}$$

where $BigM$ is a sufficiently large value that resets the accumulated waste (w_{it}) at the collection point i when visited. For example, in Eq. (3a), if the collection point i is not visited the previous day, $t-1$, $\sum_{j\in I^0}\sum_{v\in V} x_{ijv(t-1)} = 0$ and Eq. (3a) is reduced to $w_{it} \geq W_i + w_{i(t-1)}$. Conversely, when the collection point is visited the previous day, $t-1$, $\sum_{j\in I^0}\sum_{v\in V} x_{ijv(t-1)} = 1$ and Eq. (3a) is reduced to $w_{it} \geq 0$, since it is a non-negative continuous variable. This is why it is necessary to add Eq. (3c). Note that, in either case, the value of w_{it} will be pushed to the minimum ($w_{it} = W_i + w_{i(t-1)}$ or $w_{it} = W_i$, respectively) due to the sense of the optimization. A similar behavior is observed in Eq. (3b). After careful consideration, $BigM$ was set as $BigM = \max_{b\in B}\{CAP_b\}$.

Usually, linear models have certain advantages over quadratic models, which tend to be more mature than their non-linear counterparts (Rodriguez & Vecchietti, 2013). Taking this into account, a MILP model was also implemented by applying the linearization technique proposed by Glover (1975). Thus, to linearize TT_{vt}, Eqs. (4a)–(4c) are added to the model incorporating the non-negative continuous variable z_{ibjvt}.

$$z_{ibjvt} \leq n_{bj},\ \forall i, j \in I, b \in B, v \in V, t \in T \tag{4a}$$

$$z_{ibjvt} \leq x_{ijvt},\ \forall i, j \in I, b \in B, v \in V, t \in T \tag{4b}$$

$$z_{ibjvt} \geq n_{bj} + x_{ijvt} - 1,\ \forall i, j \in I, b \in B, v \in V, t \in T \tag{4c}$$

Thus, TT_{vt} can be rewritten as the linear expression of Eq. (5).

$$TT'_{vt} = T_U \sum_{i\in I} x_{i0vt} + \sum_{i\in I^0}\sum_{j\in I^0}\left(C_{ij}x_{ijvt} + \sum_{b\in B} S_b z_{ibjvt}\right) \tag{5}$$

When TT_{vt} is replaced by TT'_{vt} in the objective function (2) and restriction (2h) the model becomes linear. For showing the effectiveness of this linearization, both formulations (MIQP and MILP) were solved. Specifically, they were coded using pyomo (Bynum et al., 2021) and solved with Gurobi v10.0.2 (Gurobi Optimization, 2023).

Moreover, unlike Rossit et al. (2024), two valid cuts were incorporated into the formulation to break the symmetry between solutions. The first cut ensures that a vehicle with index l can only starts its route if the vehicle with index $l-1$ has already departed. Without this constraint, unused vehicles could replace those in operation, resulting in symmetric solutions. This is prevented by Eq. (6a). The second cut mandates that vehicles start their routes with empty loads. This eliminates solutions where vehicles start with partial loads, preventing alternate plans that utilize remaining capacity differently. Eq. (6b) addresses this issue. Both valid cuts, originally proposed by Dror et al. (1994) for the periodic routing problem, are applied here for the periodic allocation-routing problem.

$$\sum_{\substack{j\in I\\ j\neq 0}} x_{0jl} \leq \sum_{\substack{j\in I\\ j\neq 0}} x_{0jl-1},\ \forall\, l \in L, l > 0. \tag{6a}$$

$$\sum_{\substack{j\in I\\ j\neq 0}} v_{0jl} = 0,\ \forall\, l \in L. \tag{6b}$$

4.2 Metaheuristic methods

Metaheuristic procedures are a class of approximate methods that are designed to solve difficult combinatorial optimization problems where classical heuristics are not effective (Osman and Kelly, 1996). Here, a single solution-based metaheuristic and a population-based metaheuristic were implemented to solve the mathematical model described in Section 3.

In both cases, two chromosomes are considered to encode the candidate solutions: $x \to (pop, mask)$ being $pop = [p_{it}]$ and $mask = [m_{it}]$ with $i \in I$ and $t \in (T - T')$, where pop is a matrix whose rows correspond to the daily permutations of the order of MSW collection from the n_I collection points, assuming that all of them are visited every day, and $mask$ is a binary matrix whose terms m_{it} take the value 1, if on day $t \in (T - T')$ MSW is collected from collection point i, or 0 otherwise. This definition of $mask$ guarantees that restriction (2e) is satisfied. Furthermore, by handling permutations, constraints (2d) and (2f) are always satisfied. Also, by not incorporating the depot in the permutations, it facilitates the constraint (2a) to be fulfilled. The remaining constraints are incorporated into the GA when the fitness function is evaluated. On the one hand, individuals are repaired by adding ones to the $mask$ when $t \in (T - T')$, in order to meet constraints (2b), (2c) and (2i) – (2m). On the other hand, they are penalized in case they do not meet any of the restrictions (2g) and (2h), i.e. if the number of daily routes exceeds the size of the vehicle fleet (n_V) and/or if a route cannot be completed within the drivers' working day, as shown in Eq. (7).

$$f_{fitness}(x) = f(x) + \lambda \sum_{\substack{t \in (T - T') \\ nr_t > n_V}} (nr_t - n_V)/n_V + \gamma \sum_{\substack{v \in V \\ t \in (T - T') \\ TT_{vt} > T_L}} (TT_{vt} - T_L) \tag{7}$$

where $f(x)$ is the objective function (2), nr_t is the number of routes on day $t \in (T - T')$, $\lambda \in \mathbb{R}^+$ and $\gamma \in \mathbb{R}^+$.

The rest of variables that appear in the mathematical formulation (y, w, w^{max} and n) are determined when the fitness function is calculated.

As an illustrative example, the chromosomes associated with the solution obtained for the i.12.1 instance with the MILP model is presented in the Appendix A.

4.2.1 Genetic Algorithm

Genetic Algorithm is a population-based metaheuristic method proposed by Holland (1992) that is inspired by Charles Darwin's natural evolution and was designed to search the optimal solution in optimization problems. The basic elements of GA are the representation of the problem solutions and operators inspired by biological evolution: selection (of the individuals best adapted to the environment, i.e. to the problem), crossover or recombination and mutation. Selection is independent of representation, but the crossover and mutation operators are not.

As shown in Figure 1, evolution in GA usually starts from a population of individuals, or set of potential solutions to the problem, generated randomly, and continues with an iterative process, in which each new population is associated with a new generation. In each generation, the fitness of each individual in the population is evaluated and, depending on this value, candidate individuals are selected to be recombined, with a certain crossover rate, and whose descendants can be mutated or not, with a certain mutation rate, to give rise to a new generation. To avoid the loss of the best solution(s) found in a given generation, elitist

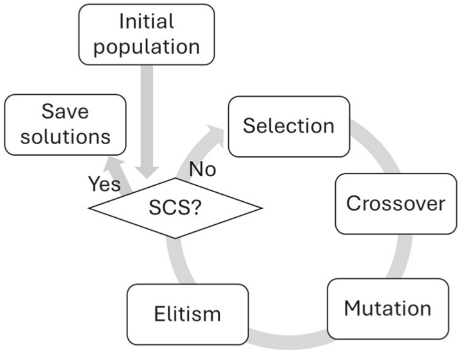

Fig. 1 Genetic Algorithm flowchart

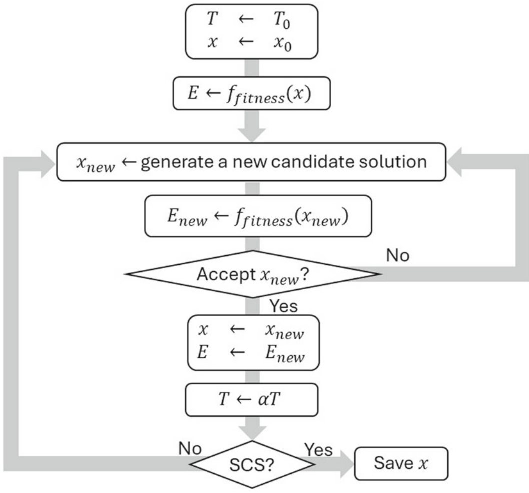

Fig. 2 Simulated Annealing algorithm flowchart

strategies are usually applied that may or may not copy these elitist individuals in the next generation.The algorithm terminates when a stopping criterion is satisfied (SCS).

4.2.2 Simulated Annealing

Simulated Annealing is a single-solution-based metaheuristic method that was introduced in combinatorial optimization by Kirkpatrick, Gelatt and Vecchi in the early 1980s (Kirkpatrick et al., 1983). It simulates the annealing process in metallurgy to approximate the global minimum of an optimization problem. As shown in Figure 2, the process of finding the global optimum starts with a high energy state (an initial solution, x_0, at an initial temperature, T_0) and the temperature, T, is gradually reduced (with a cooling factor, α) until it reaches a state of minimum energy (the optimal solution) or reaches a maximum number of evaluations of the fitness function. The solution update method depends on the representation of the candidate solutions. The acceptance criterion for new solutions is based on the Metropolis algorithm (Metropolis et al., 1953) in which a new solution, x_{new}, is accepted if it has lower energy, i.e., $\Delta E = E_{new} - E < 0$, or even if it has the same or higher energy, it is accepted with a certain probability that depends on the temperature: $P(\Delta E) = \exp(-\Delta E/T)$.

The initial temperature plays an important role in simulated annealing. In fact, for simulated annealing to find good solutions, this parameter must be carefully calculated. In this paper, the initial temperature is approximated by assuming cooling in polynomial time (Delahaye et al., 2019). In this way, a given number m of transitions are randomly generated and the number of transitions that strictly improve the value of the objective function, m_1, is counted. Let $\triangle_f^-(+)$ be the average of the objective function differences between all increasing transitions, T_0 is then calculated from equation Eq. (8), where $\chi(T)$ is a user-defined acceptance rate.

$$T_0 \simeq \frac{\triangle_f^-(+)}{\ln\left(\frac{m-m_1}{(m-m_1)\chi(T)-m_1(1-\chi(T))}\right)} \tag{8}$$

5 Computational experimentation

This Section reports the instances used in the paper as well as the computational experimentation performed to tune the metaheuristic methods proposed. For the parameterization, respective design of experiments were performed on the $i.12.1$ instance, involving different factors (parameters and operators or methods) at given levels. In what follows, we will understand that a combination of levels of the different factors involved in a given design of experiments is a treatment. In all cases, 30 independent runs of each treatment were performed on an Intel(R) Core(TM) i7-10700K processor with 16 GB of RAM.

The objective of each design of experiments is to study the effect of the factors on the performance and runtime of each metaheuristic when searching for an optimal or quasi-optimal solution of the considered problem. So that the penalties for infeasible solutions in the fitness function defined by Eq. (7) would not be too high, we set $\lambda = 100$ and $\gamma = 1000$. In order to analyze the results of each experimental design, box-and-whisker plots are plotted and a Kruskal-Wallis Rank Sum Test (Hollander & Wolfe, 1973) has been carried out to establish whether there are significant differences between the levels of each factor. When the value of a Kruskal-Wallis test is significant (p-value < 0.05), a multiple comparison test after Kruskal-Wallis (Siegel & Castellan, 1988) between treatments has been carried out to identify which levels are different with pairwise comparisons adjusted appropriately for multiple comparisons: if TRUE (T), then statistically significant differences are found between the compared levels. In this case, pairs of levels were compared by applying the Mann-Whitney test (Hollander & Wolfe, 1973). All test were performed in RStudio Desktop (Posit team, 2023), a coding environment for R (R Core Team, 2022), a free software environment for statistical computing and graphics.

5.1 Description of instances

This paper presents results obtained with several instances based on data collected in field studies in the city of Bahía Blanca, Argentina (Cavallin et al., 2020). The instances, which can be downloaded from Github[1], have the generic name $i.n_I.m$, where n_I represents the number of collection points and m is a numbering of the instances in each particular case. Specifically, $m = 1, \ldots, 5$ when $n_I = 12$, $m = 1, 2, 3$ when $n_I = 15$ and $m = 1$ when $n_I \in \{40, 80, 120, 163\}$. For example, there is a single instance of 163 collection points whose

[1] https://github.com/diegorossit/ANOR-S-24-01950.git

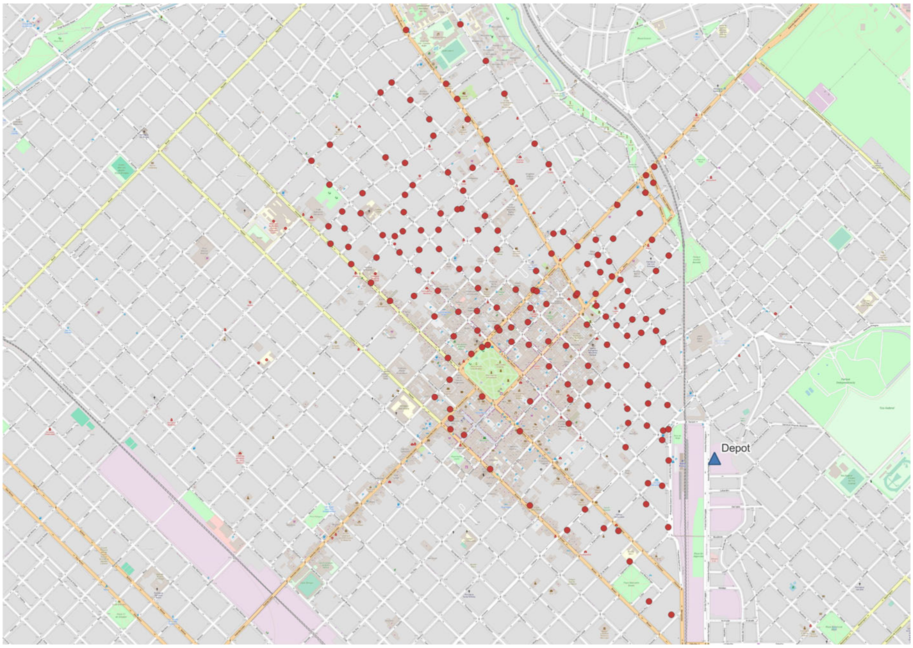

Fig. 3 Collection points (in red) of instance $i.163.1$ (Bahía Blanca, Argentina). The blue triangle marks the location of the depot

name is $i.163.1$. Each instance consists of two files: *time.txt*, which contains the matrix of travel times (minutes) from the depot to the collection points, between collection points, and from the collection points to the depot, and *waste.txt*, which contains the geographic coordinates of each collection point, as well as the waste (m^3) generated daily by the urban area assigned to each collection point. Figure 3 shows the 163 collection points of instance $i.163.1$. The smaller instances were randomly generated from this one. As an illustrative case, the daily routes obtained in the solution of the MILP model on instance $i.12.1$ are presented in the Appendix A.

Table 2 shows the capacity, service time and weekly cost associated with the eight bin combinations considered in this study, which along with the specifications of the three types of bins that compose them, can be found in Mahéo et al. (2023). However, unlike their approach, which assumes a uniform service time for all bin combinations, in this work the service time has been estimated based on the specific type of the bins constituting each combination. This modification adds more complexity to the objective problem and causes the appearance of bilinear terms in Eq. (2). The estimated lifespan of the bis is ten years (Brogaard & Christensen, 2012) and the maintenance cost of each bin is approximately 5% of the purchase cost (D'Onza et al., 2016). To calculate the estimated weekly cost of each bin combination, the purchase and maintenance expenses for each bin were summed and divided by the total number of weeks of expected lifespan. Bins service times are estimated using the field work of Carlos et al. (2019), who considered the collection vehicle fleet to be homogeneous.

As in Mahéo et al. (2023), since the instances are smaller than the actual collection zones in the city, vehicle capacity, fleet size and working day length are adjusted so that the problem does not become trivial, where a single vehicle can collect all the waste in a single trip. In this sense, a vehicle capacity of 12 and 15 m^3 have been considered for the instances of 12

Table 2 Description of the bin combinations considered

Bin combination	Capacity (m^3)	Service time (minutes)	Weekly cost (US$)
0	1.1	0.70	0.78
1	2.2	1.40	1.56
2	2.4	0.66	1.56
3	3.3	2.10	1.56
4	3.5	1.36	1.56
5	4.3	1.37	1.56
6	4.8	1.32	1.56
7	5.6	1.33	1.56

and 15 collection points, respectively, and 21 m^3 for the rest. Moreover, and in contrast to that work, this paper considers a longer planning horizon taking into account a whole week (seven days), with no waste collection scheduled on Sundays as this is a rest day for drivers.

Finally, the estimated cost of the collection vehicles was set at US$0.5764 per minute (D'Onza et al., 2016), and based on the paper of Owusu et al. (2019) a vehicle unloading time of 8 minutes was considered. In addition, since the mathematical model establishes that each vehicle can perform at most one route per day, the size of the fleet and the duration of the working day has been established by applying Eqs. (9) and (10), respectively.

$$n_V = \lceil n_I/10 \rceil = \min\{k \in \mathbb{Z} | n_I/10 \leq k\} \tag{9}$$

$$T_L = \lceil \sum_{i,j \in I_0} C_{ij} / \big(n_V(n_V - 1)|T - T'|\big) \rceil \tag{10}$$

5.2 GA parametrization

A 4×3^3 factorial design on the i.12.1 instance was considered to tune the mixed GA, specifically focusing on the genetic operators associated to the representation with permutations. A total of 3,240 runs of the mixed GA were performed. The factors of this factorial design are: *Crossover operator* (F1), *Crossover rate* (F2), *Mutation operator* (F3) and *Mutation rate* (F4). Their respective levels are:

F1: (1) Partially Mapped Crossover (PMX) (Goldberg & Lingle, 2014), (2) Order Crossover (OX) (Davis, 1985), (3) Cicle Crossover (CX) (Oliver et al., 1987) and (4) Modified Cicle Crossover (CX2) (Hussain et al., 2017).
F2: Crossover rate: (1) 0.8, (2) 0.85 and (3) 0.9.
F3: Mutation operator: (1) Exchange mutation (EM), (2) Insertion mutation (IM) and (3) Inversion mutation (INM) (Rossit et al., 2024).
F4: Mutation rate: (1) 0.05, (2) 0.10 and (3) 0.15.

As can be seen in Algorithm 1, during each successive generation, individuals are selected from the current population to produce a new generation. In this paper, a tournament selection that randomly samples two individuals without replacement and selects the one with the best fitness function value was considered and, for binary representation, two-point crossover and uniform mutation operators were chosen, as they have proven to be efficient operators in the literature. In addition, populations of 100 individuals with elitism of 2 and 1000 generations were fixed. For binary encoding, the crossover rate is the same as permutation encoding and the mutation rate is equal to 1/number of collection points.

Algorithm 1 Mixed GA

Require: problem data and algorithm parameters: population size (N), number of generations ($Ngen$), selection, crossover and mutation operators, crossover and mutation rates and number of elite solutions (individuals) considered (≥ 0).
Generate a mixed initial population: $P^0 \leftarrow \{(pop, mask)_n | 0 \leq n < N\}$
Evaluate the fitness function of each individual of the population.
Save the elitist solutions.
$ng = 0$
while $ng < Ngen$ **do**
  $ng = ng + 1$
  Select individuals to be crossed (2 times the population size).
  for i in $range(0, N, 2)$ **do**
    Cross consecutive pairs (of both encodings) of the selected individuals, with the fixed crossover rate.
    Mute the descendants (of both encodings) of the previous crossover of two parents, with the fixed mutation rate.
    Evaluate the fitness function of the mutated descendants.
    Insert the mutated descendants into the new generation.
  end for
  Apply elitism by adding the elitist solutions obtained in the previous generation to the new generation.
  Save the elitist solutions of the new generation.
end while
Save the best individual found in a file.

Fig. 4 shows the box-and-whisker plots associated with the overall cost results grouped by each factor. Table 3 shows the output of the different hypothesis tests performed. From the p-values obtained when performing a Kruskal-Wallis rank sum test on the overall costs, it can be concluded that, for each factor, there are at least a pair of levels that are significantly different. Also, it can be seen that the multiple comparison test after Kruskal-Wallis detected significant differences between all pairs of levels of factors F1 and F2, but only detected significant differences between pair 1–3 of factor F3 and pairs 1–2 and 1–3 of factor F4. That is, box-and-whisker plots and hypothesis tests support that the best performing treatment is: (F1 = CX, F2 = 0.8, F3 = EM, F4 = 0.05), but also (F1 = CX, F2 = 0.8, F3 = IM, F4 = 0.05). However, the Mann-Whitney test detected significant differences between this two treatments and establishes that the former is shifted to the left of the latter.

Box-and-whisker plots, when the response variable is the mixed GA runtime (seconds), is shown in Figure 5. Table 4 shows the p-values obtained when performing a Kruskal-Wallis rank sum test on the mixed GA runtime. It can be concluded that, for each factor, there are at least a pair of levels that are significantly different. Table 4 also shows that the multiple comparison test after Kruskal-Wallis detected significant differences between all pairs of levels of factors F1, F2 and F4, but only detected significant differences between pair 1–3 of factor F3. It can be concluded that the hypothesis tests support that the crossover operator that needs the least runtime is CX2 closely followed by CX (see Figure 5). However, the performance of CX is significantly better than that of CX2 (see Figure 4). Furthermore, it can be concluded that the lower the crossover rate and the lower the mutation rate, the shorter the runtime. Regarding the mutation operators, it can be concluded that the lowest runtimes correspond to EM and IM.

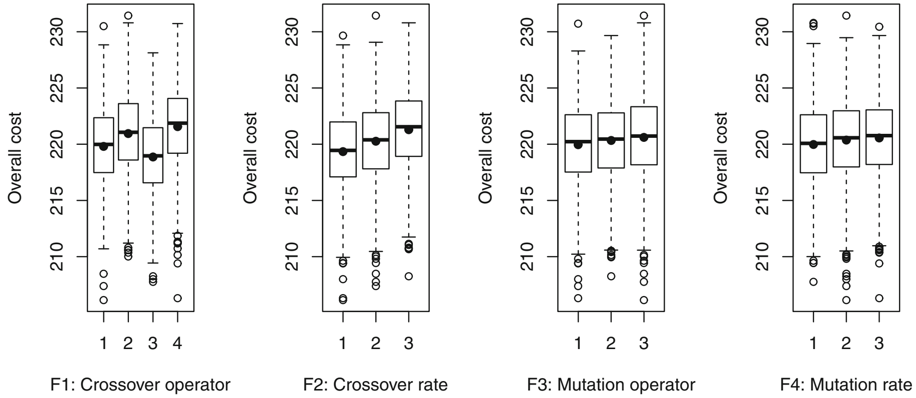

Fig. 4 GA parametrization. Box-and-whisker plots when the response variable is the overall cost (US$)

Table 3 GA parametrization. Overall cost: summary of statistical tests

Kruskal-Wallis rank sum test: p-value

F1: Crossover operator	F2: Crossover rate	F3: Mutation operator	F4: Mutation rate
$< 2.2e-16$	$< 2.2e-16$	0.0004956	0.0005319

The multiple comparison test after Kruskal-Wallis

F1: Crossover operator		F2: Crossover rate	F3: Mutation operator	F4: Mutation rate
1–2 T	2–3 T	1–2 T	1–2 F	1–2 T
1–3 T	2–4 T	1–3 T	1–3 T	1–3 T
1–4 T	3–4 T	2–3 T	2–3 F	2–3 F

Mann-Whitney test: alternative hypothesis and p-values

F1: Crossover operator		F2: Crossover rate		F3: Mutation operator		F4: Mutation rate	
$3 < 1$	$3.219e-07$	$1 < 2$	$8.208e-09$	$1 < 2$	0.01747	$1 < 2$	0.004533
$3 < 2$	$< 2.2e-16$	$1 < 3$	$< 2.2e-16$	$1 < 3$	$5.228e-05$	$1 < 3$	$8.013e-05$
$3 < 4$	$< 2.2e-16$						

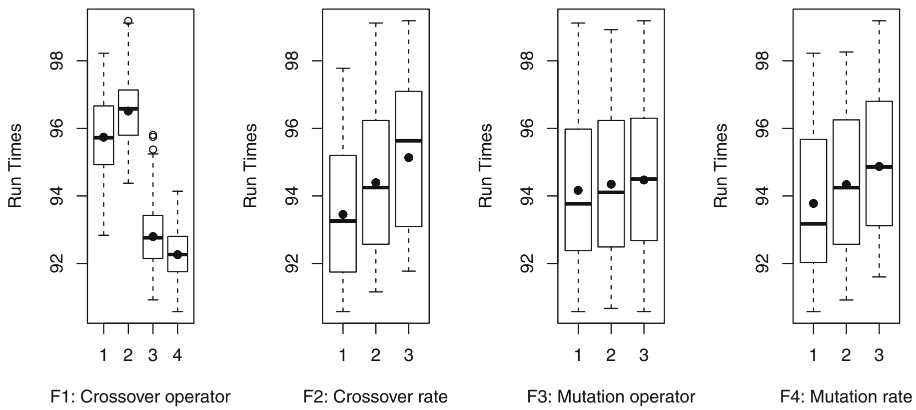

Fig. 5 Box-and-whisker plots when the response variable is the mixed GA runtime (seconds)

Table 4 Mixed GA runtime: summary of statistical tests

Kruskal-Wallis rank sum test: p-value

F1: Crossover operator	F2: Crossover rate	F3: Mutation operator	F4: Mutation rate
$< 2.2e-16$	$< 2.2e-16$	0.0007906	$< 2.2e-16$

The multiple comparison test after Kruskal-Wallis

F1: Crossover operator		F2: Crossover rate	F3: Mutation operator	F4: Mutation rate
1–2 T	2–3 T	1–2 T	1–2 F	1–2 T
1–3 T	2–4 T	1–3 T	1–3 T	1–3 T
1–4 T	3–4 T	2–3 T	2–3 F	2–3 T

Mann-Whitney test: alternative hypothesis and p-values

F1: Crossover operator		F2: Crossover rate		F3: Mutation operator		F4: Mutation rate	
$4 < 1$	$< 2.2e-16$	$1 < 2$	$< 2.2e-16$	$1 < 2$	0.014	$1 < 2$	$1.09e-13$
$4 < 2$	$< 2.2e-16$	$1 < 3$	$< 2.2e-16$	$1 < 3$	$8.511e-05$	$1 < 3$	$< 2.2e-16$
$4 < 3$	$< 2.2e-16$						

Table 5 Mixed SA: thousands of fitness function evaluations required for each treatment to achieve the final temperature

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
EM	80	115	150	200	260	355	505	805	1, 705
IM	80	115	150	200	260	355	505	805	1, 705
INM	80	115	150	200	260	355	505	805	1, 705

5.3 SA parametrization

A 3×9 factorial design on the $i.12.1$ instance was considered to tune the mixed SA, specifically focusing on the solution update method associated to the representation with permutations and the cooling factor. For binary representation, uniform mutation operators was chosen as the solution update method. A total of 810 runs of the mixed SA were performed. The factors of this factorial design are: *Solution update method* (F1) and *Cooling factor* (F2). Their respective levels are:

- **F1**: Solution update method: (1) EM, (2) IM and (3) INM.
- **F2**: Cooling factor: (1) 0.1, (2) 0.2, (3) 0.3, (4) 0.4, (5) 0.5, (6) 0.6, (7) 0.7, (8) 0.8 and (9) 0.9.

For all instances, an acceptance rate of 0.8 and a maximum iteration number at each temperature of 5,000, with a maximum number of 100 repetitions of the best value of the objective function, have been set. The initial temperature was approximated for each solution update method by applying Eq. (8), obtaining values of 3,833 (EM), 3,890 (IM) and 3,847 (INM). The final temperature was fixed at $1.e-06$ (or less). Table 5 shows the number of fitness function evaluations required for each treatment to achieve the final temperature. As expected, it can be seen that the slower the cooling, the more fitness function evaluations are needed.

Fig. 6 shows the box-and-whisker plots associated with the overall cost results grouped by each factor. Table 6 shows the output of the different hypothesis tests performed. From the p-values obtained when performing a Kruskal-Wallis rank sum test on the overall costs, it can be

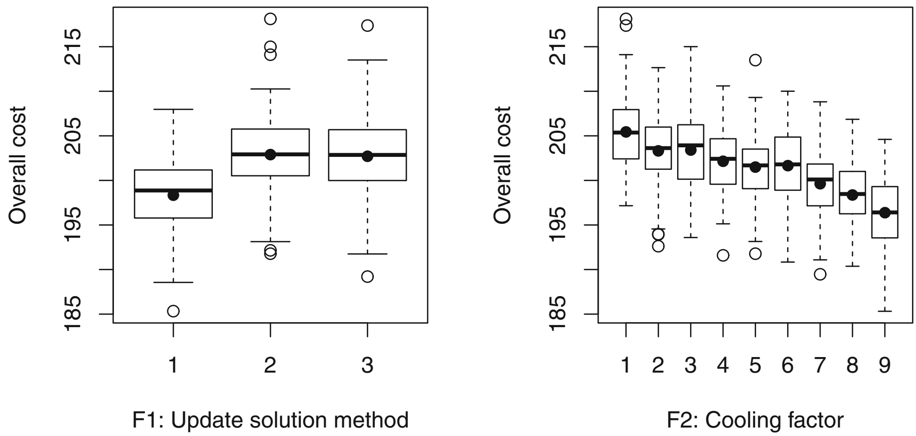

Fig. 6 SA parametrization. Box-and-whisker plots when the response variable is the overall cost (US$)

concluded that, for each factor, there are at least a pair of levels that are significantly different. Also, it can be seen that the multiple comparison test after Kruskal-Wallis detected significant differences between pairs 1–2 and 1–3 of factor F1 and detected significant differences between the cooling factors 0.8 and 0.9 and all the others, but not between them. However, the Mann-Whitney test detected that the cooling factors 0.9 performs better than all the others, including 0.8. That is, the box-and-whisker plots and the hypothesis tests support that the best performing treatment is: (F1 = EM, F2 = 0.9).

Box-and-whisker plots, when the response variable is the mixed SA runtime (seconds), is shown in Figure 7. Table 7 shows the p-values obtained when performing a Kruskal-Wallis rank sum test on the mixed SA runtime. It can be concluded that, for each factor, there are at least a pair of levels that are significantly different. Table 7 also shows that the multiple comparison test after Kruskal-Wallis detected significant differences between all pairs of levels of factor F1, with level 1 requiring the least runtime. With regard to the F2 factor, no significant differences are found between levels 2, 3 and 4, which have a lower computational cost than all the others. However, the Mann-Whitney test detected that cooling factor 0.2 needs less runtime than all others, including 0.3 and 0.4. In any case, the performance of levels 2, 3 and 4 is worse than that of levels 7, 8 and 9 (see Figure 6 and Table 6). In general, the lower the cooling, the more evaluations of the fitness function are needed and, therefore, the higher the computational cost, but also the higher the performance of the SA algorithm.

6 Results and discussion

This section presents the results obtained both when comparing the exact models (MIQP and MILP) with the metaheuristics considered (mixed GA and mixed SA) - with the five instances of 12 collection points and the three instances of 15 collection points - and when comparing the two metaheuristics with the remaining instances considered. Small instances involve a total of 2,366 (12 collection points) and 3,584 (15 collection points) binary variables, which in the case of metaheuristics translates into two chromosomes of size 72 and 90, respectively. Larger instances did not produce feasible solutions with the exact models due to the large number of binary variables involved (between 47,068, in the case of 40 collection points,

Table 6 SA parametrization. Overall cost: summary of statistical tests

Kruskal-Wallis rank sum test: p-value

F1: Solution update method	F2: Cooling factor
$< 2.2e - 16$	$< 2.2e - 16$

The multiple comparison test after Kruskal-Wallis

F1: Solution update method	F2: Cooling factor					
1–2 T	1–2 F	1–8 T	2–7 T	3–7 T	4–8 T	6–7 F
1–3 T	1–3 F	1–9 T	2–8 T	3–8 T	4–9 T	6–8 T
2–3 F	1–4 T	2–3 F	2–9 T	3–9 T	5–6 F	6–9 T
	1–5 T	2–4 F	3–4 F	4–5 F	5–7 F	7–8 F
	1–6 T	2–5 F	3–5 F	4–6 F	5–8 T	7–9 T
	1–7 T	2–6 F	3–6 F	4–7 T	5–9 T	8–9 F

Mann-Whitney test: alternative hypothesis and p-values

F1: Solution update method		F2: Cooling factor					
$1 < 2$	$< 2.2e - 16$	$1 > 4$	$9.381e - 08$	$2 > 8$	$1.186e - 14$	$4 > 9$	$< 2.2e - 16$
$1 < 3$	$< 2.2e - 16$	$1 > 5$	$1.882e - 11$	$2 > 9$	$< 2.2e - 16$	$5 > 8$	$4.296e - 08$
		$1 > 6$	$2.125e - 08$	$3 > 7$	$6.076e - 09$	$5 > 9$	$1.21e - 15$
		$1 > 7$	$< 2.2e - 16$	$3 > 8$	$3.02e - 13$	$6 > 8$	$1.728e - 07$
		$1 > 8$	$< 2.2e - 16$	$3 > 9$	$< 2.2e - 16$	$6 > 9$	$1.004e - 13$
		$1 > 9$	$< 2.2e - 16$	$4 > 7$	$1.908e - 05$	$7 > 9$	$1.419e - 07$
		$2 > 7$	$1.14e - 09$	$4 > 8$	$8.987e - 10$	$8 > 9$	0.0005458

Table 7 Mixed SA runtime: summary of statistical tests

Kruskal-Wallis rank sum test: p-value

F1: Solution update method	F2: Cooling factor
$< 2.2e-16$	$< 2.2e-16$

The multiple comparison test after Kruskal-Wallis

F1: Solution update method	F2: Cooling factor					
1–2 T	1–2 T	1–8 T	2–7 T	3–7 T	4–8 T	6–7 T
1–3 T	1–3 F	1–9 T	2–8 T	3–8 T	4–9 T	6–8 T
2–3 T	1–4 F	2–3 F	2–9 T	3–9 T	5–6 F	6–9 T
	1–5 F	2–4 F	3–4 F	4–5 F	5–7 T	7–8 F
	1–6 T	2–5 T	3–5 T	4–6 T	5–8 T	7–9 F
	1–7 T	2–6 T	3–6 T	4–7 T	5–9 T	8–9 F

Mann-Whitney test: alternative hypothesis and p-values

F1: Solution update method		F2: Cooling factor					
$1 < 2$	$< 2.2e-16$	$1 > 2$	$3.116e-05$	$2 < 7$	$< 2.2e-16$	$4 < 7$	$< 2.2e-16$
$1 < 3$	$< 2.2e-16$	$1 < 6$	$1.485e-07$	$2 < 8$	$< 2.2e-16$	$4 < 8$	$< 2.2e-16$
$2 < 3$	0.0005991	$1 < 7$	$< 2.2e-16$	$2 < 9$	$< 2.2e-16$	$4 < 9$	$< 2.2e-16$
		$1 < 8$	$< 2.2e-16$	$3 < 5$	$1.721e-08$	$5 < 7$	$< 2.2e-16$
		$1 < 9$	$< 2.2e-16$	$3 < 6$	$< 2.2e-16$	$5 < 8$	$< 2.2e-16$
		$2 < 3$	0.02207	$3 < 7$	$< 2.2e-16$	$5 < 9$	$< 2.2e-16$
		$2 < 4$	$1.749e-05$	$3 < 8$	$< 2.2e-16$	$6 < 7$	$2.714e-11$
		$2 < 5$	$1.03e-11$	$3 < 9$	$< 2.2e-16$	$6 < 8$	$< 2.2e-16$
		$2 < 6$	$< 2.2e-16$	$4 < 6$	$1.017e-09$	$6 < 9$	$< 2.2e-16$

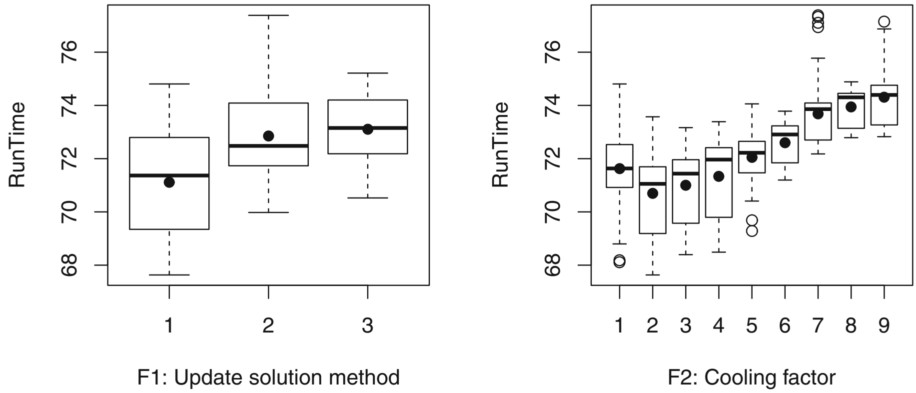

Fig. 7 Box-and-whisker plots when the response variable is the mixed SA runtime (seconds)

Table 8 Small instances: Statistical summary of the overall costs over 30 independent mixed GA runs

Instance	Min	Median	Mean	Confidence interval		Runtime (seconds)	S-W test
i.12.1	194.6	203.4	202.7	201.5	203.9	1,686.1	0.3064
i.12.2	196.8	202.3	201.9	201.0	202.9	951.9	0.6309
i.12.3	198.8	204.1	204.2	203.3	205.1	949.6	0.7766
i.12.4	189.9	193.9	194.2	193.3	195.0	1,970.6	0.8247
i.12.5	189.4	196.8	196.6	195.5	197.7	2,007.1	0.9716
i.15.1	220.2	225.4	225.9	224.6	227.2	2,376.6	0.0999
i.15.2	212.9	222.5	222.2	220.6	223.6	2,397.3	0.8849
i.15.3	234.7	238.2	238.9	237.6	239.7	2,367.1	0.0046

and 3,200,624, in the case of 163 collection points). In view of the conclusions made in the previous section, all mixed GA runs were performed with treatment: (F1 = CX, F2 = 0.8, F3 = EM, F4 = 0.05) and all mixed SA runs were performed with treatment: (F1 = EM, F2 = 0.9). In both cases, the other parameters considered were the same as those of the respective tunings, but this time the stopping criterion for each run of the GA was set to the number of fitness function evaluations required for the SA to reach a final temperature set at $1.e - 12$ (or less). All instances were run on a Processor 13th Gen Intel(R) Core(TM) i9-13900K with 128 GB of RAM.

Tables 8 and 9 summarize the statistical results obtained from a sample of 30 independent runs for the five instances of 12 collection points and the three instances of 15 collection points, with GA and SA, respectively. From left to right, Tables 8 and 9 show, for each sample of the 30 best overall costs (US$) obtained with the 30 independent runs: the minimum, median, mean, 95 percent confidence interval of the mean or of the (pseudo)median, as well as the mean run time (in seconds) and the p-value obtained in the Shapiro Wilk (S-W) test with the **shapiro.test()** function. If a distribution is symmetric, then the pseudomedian and median coincide (Hollander & Wolfe, 1973). When the value of a S-W test is significant (p-value < 0.05) then the distribution of the sample is significantly different from a normal distribution. Based on this p-value, a 95 percent confidence interval is calculated for the (pseudo)median with the **wilcox.test()** function, if p-value < 0.05, or for the mean with

Table 9 Small instances: Statistical summary of the overall costs over 30 independent mixed SA runs

Instance	Min	Median	Mean	Confidence interval		Runtime (seconds)	S-W test	T_0	FFE
$i.12.1$	188.0	193.1	193.7	192.7	194.7	1,357.2	0.3985	3,833	1.705e06
$i.12.2$	188.9	193.6	193.6	192.6	194.7	1,356.6	0.6987	2,600	1.685e06
$i.12.3$	192.3	197.3	197.2	196.3	198.0	1,308.7	0.4580	2,089	1.675e06
$i.12.4$	179.2	186.6	185.8	184.5	187.1	1,353.6	0.2715	3,259	1.700e06
$i.12.5$	185.1	188.4	188.4	187.5	189.3	1,379.7	0.2054	3,856	1.705e06
$i.15.1$	210.6	214.4	214.8	213.8	215.6	1,572.8	0.8231	2,250	1.680e06
$i.15.2$	206.5	210.5	210.8	210.1	211.6	1,676.1	0.9116	2,370	1.685e06
$i.15.3$	226.0	228.7	229.0	228.3	229.6	1,641.1	0.3495	2,160	1.680e06

Table 10 Small instances: MILP and MIQP solutions

Instance	Bin cost	Routing cost	Overall cost	Lower bound	Optimality gap
MILP					
$i.12.1$	36.93	141.49	178.42	83.92	52.96 %
$i.12.2$	39.95	123.76	163.71	86.84	46.95 %
$i.12.3$	42.85	124.13	166.98	95.56	42.77 %
$i.12.4$	45.74	111.95	157.70	88.85	43.65 %
$i.12.5$	37.30	128.71	166.01	84.06	49.36 %
$i.15.1$	57.40	115.58	172.98	124.66	27.93 %
$i.15.2$	59.87	108.57	168.43	105.52	33.94 %
$i.15.3$	63.91	125.17	189.08	136.17	28.01 %
MIQP					
$i.12.1$	39.58	117.32	156.90	81.94	53.96 %
$i.12.2$	36.53	150.05	186.58	85.81	46.95 %
$i.12.3$	42.48	120.85	163.33	86.54	42.77 %
$i.12.4$	43.53	118.07	161.60	86.31	43.66 %
$i.12.5$	38.44	131.20	169.64	84.34	49.36 %
$i.15.1$	47.32	137.90	185.22	99.70	46.17 %
$i.15.2$	44.76	154.54	203.02	95.61	50.97 %
$i.15.3$	64.99	131.83	196.82	110.04	44.09 %

the **t.test()** function, otherwise. These three test functions can be found in the R package *stats* (R Core Team, 2022). Table 9 also shows the initial temperature, T_0, and the number of fitness function evaluations, FFE, for each instance. Precisely this number of fitness function evaluations was used as the stopping criterion for GA in order to compare the performance of both algorithms.

Table 10 shows the results obtained when solving the instances of 12 and 15 collection points with the MILP and MIQP models (whose running times were set at 8 hours). From left to right, this table shows the bin installation and maintenance cost, the MSW collection cost, the overall cost, the lower bound estimated by Gurobi (dual objective bound) and the

Table 11 Small instances: MILP versus MIQP

Cost	$i.12.1$	$i.12.2$	$i.12.3$	$i.12.4$	$i.12.5$	$i.15.1$	$i.15.2$	$i.15.3$
Bin (C_B)	0.07	−0.09	−0.01	−0.05	0.03	−0.18	−0.5	0.02
Routing (C_R)	−0.17	0.21	−0.03	0.05	0.02	0.19	0.42	0.05
Overall (C_O)	−0.12	0.14	−0.02	0.02	0.02	0.07	0.18	0.04

optimality gap reported by Gurobi, which is calculated with Eq. (11).

$$\text{Optimality gap} = \frac{|\text{Overall cost} - \text{lower bound}|}{|\text{Overall cost}|} \tag{11}$$

The optimality gap is useful in computationally complex problem, such as the optimization problem addressed in this paper, where it is usual for exact solvers to report feasible but not proven optimal solutions. In these cases, this gap, which is estimated with the lower bound, indicates an estimate of how far the obtained feasible solution might be from the optimal solution. Further details on how Gurobi computes the optimality gap can be found at Gurobi Optimization (2023).

The comparison between the results of Gurobi are shown in Table 11, where the percentage differences between the costs obtained with MILP and MIQP has been calculated with Eq. (12).

$$\%\ \text{dif. } C_* = 100\,(C_*^{MIQP} - C_*^{MILP})/C_*^{MILP} \tag{12}$$

where C_*^{MIQP} is the cost obtained with MIQP and C_*^{MILP} is the cost obtained with MILP. It can be seen that a positive value of % dif. C_* means that the MILP formulation obtained a better solution than MIQP.

Therefore, looking at Table 11, it can be concluded that, in most cases, MILP obtains solutions with better overall and routing costs than MIQP, reaching a percentage difference in overall costs of 14% with instance $i.12.2$ and 18% with instance $i.15.2$, and a percentage difference in routing costs of 21% with instance $i.12.2$ and 42% with instance $i.15.2$. Although in the case of instance $i.12.1$, MIQP obtains a better solution with a percentage difference of -12% in the overall cost and -17% in the routing cost. Furthermore, in general, MIQP obtains solutions with a lower bin cost, up to -18% with instance $i.15.1$.

The comparison of the results of Gurobi with both the best solution obtained with both metaheuristics and the corresponding mean is shown in Table 12, where the percentage difference between the costs obtained with the exact models and metaheuristics has been calculated with Eq. (13) and the percentage difference between the costs obtained with GA and SA has been calculated with Eq. (14).

$$\%\ \text{dif. } C_* = 100\,(C_*^{meta} - C_*^{exact})/C_*^{exact} \tag{13}$$

where C_*^{meta} is the cost obtained with the GA or SA algorithm and C_*^{exact} is the cost obtained with the MILP or MIQP model.

$$\%\ \text{dif. } C_* = 100\,(C_*^{GA} - C_*^{SA})/C_*^{SA} \tag{14}$$

where C_*^{GA} is the cost obtained with GA and C_*^{SA} is the cost obtained with SA.

It can be seen that, in Eq. (13), a positive value of % dif. C_* means that the exact model obtained a better solution than the metaheuristic and, in Eq. (14), it means that the SA obtained a better solution than the GA. Therefore, looking at Table 12, it can be concluded that, on the one hand, the exact models obtain better solutions than the metaheuristics considered and,

Table 12 Small instances: MILP versus MIQP, SA versus GA, and MILP and MIQP versus GA and SA

		Cost	$i.12.1$	i.12.2	$i.12.3$	$i.12.4$	$i.12.5$	$i.15.1$	$i.15.2$	$i.15.3$
	GA Min	Bin (C_B)	52.32	43.70	55.62	46.80	52.38	66.84	65.02	67.55
		Routing (C_R)	142.25	153.10	143.22	143.11	137.04	153.32	147.83	167.16
		Overall (C_O)	194.57	196.80	198.84	189.91	189.42	220.16	212.85	234.71
	GA Mean	Bin (C_B)	51.03	47.83	51.96	49.86	50.30	64.39	64.90	67.60
		Routing (C_R)	151.70	154.12	152.25	144.29	146.32	161.55	157.25	171.34
		Overall (C_O)	202.73	201.95	204.21	194.15	196.62	225.94	222.15	238.94
	SA Min	Bin (C_B)	50.87	48.71	51.27	45.35	52.32	63.57	67.55	67.55
		Routing (C_R)	137.14	140.20	141.06	133.88	132.76	147.02	138.96	158.49
		Overall (C_O)	188.01	188.91	192.33	179.23	185.08	210.59	206.51	226.04
	SA Mean	Bin (C_B)	50.04	49.16	51.17	49.68	49.47	64.36	64.89	66.98
		Routing (C_R)	143.64	144.46	146.00	136.16	138.95	150.41	145.96	162.00
		Overall (C_O)	193.68	193.62	197.17	185.84	188.42	214.77	210.85	228.98
SA vs GA	Min	% dif. C_B	2.85	−10.29	8.48	3.20	0.11	5.14	−3.75	0.00
		% dif. C_R	3.73	9.20	1.53	6.89	3.22	4.29	6.38	5.47
		% dif. C_O	3.49	4.18	3.38	5.96	2.34	4.54	3.07	3.84
	Mean	% dif. C_B	1.98	−2.71	1.54	0.36	1.68	0.05	0.02	0.93
		% dif. C_R	5.61	6.69	4.28	5.97	5.30	7.41	7.73	5.77
		% dif. C_O	4.67	4.30	3.57	4.47	4.35	5.20	5.36	4.35

Table 12 continued

		Cost	$i.12.1$	i.12.2	$i.12.3$	$i.12.4$	$i.12.5$	$i.15.1$	$i.15.2$	$i.15.3$
MILP vs GA	GA Min	% dif. C_B	41.67	9.39	29.80	2.32	40.43	16.45	8.60	5.70
		% dif. C_R	0.54	23.71	15.38	27.83	6.47	32.65	36.16	33.55
		% dif. C_O	9.05	20.21	19.08	20.43	14.10	27.27	26.37	24.13
	GA Mean	% dif. C_B	38.18	19.72	21.26	9.01	34.85	12.18	8.40	5.77
		% dif. C_R	7.22	24.53	22.65	28.89	13.68	39.77	44.84	36.89
		% dif. C_O	13.63	23.36	22.30	23.12	18.44	30.62	31.89	26.37
MIQP vs GA	GA Min	% dif. C_B	32.19	19.63	30.93	7.51	36.26	41.25	45.26	3.94
		% dif. C_R	21.25	2.03	18.51	21.21	4.45	11.18	−4.34	26.80
		% dif. C_O	24.01	5.48	21.74	17.52	11.66	18.86	6.80	19.25
	GA Mean	% dif. C_B	28.93	30.93	22.32	14.54	30.85	36.07	45.00	4.02
		% dif. C_R	29.30	2.71	25.98	22.21	11.52	17.15	1.75	29.97
		% dif. C_O	29.21	8.24	25.03	20.14	15.90	21.98	11.47	21.40
MILP vs SA	SA Min	% dif. C_B	37.75	21.93	19.65	−0.85	40.27	10.75	12.83	5.70
		% dif. C_R	−3.07	13.28	13.64	19.59	3.15	27.20	27.99	26.62
		% dif. C_O	5.37	15.39	15.18	13.66	11.49	21.74	22.60	19.55
	SA Mean	% dif. C_B	35.50	23.05	19.42	8.61	32.63	12.13	8.38	4.80
		% dif. C_R	1.52	16.73	17.62	21.63	7.96	30.13	34.44	29.42
		% dif. C_O	8.55	18.27	18.08	17.85	13.50	24.16	25.18	21.10
MIQP vs SA	SA Min	% dif. C_B	28.52	33.34	20.69	4.18	36.11	34.34	50.92	3.94
		% dif. C_R	16.89	−6.56	16.72	13.39	1.19	6.61	−10.08	20.22
		% dif. C_O	19.83	1.25	17.76	10.91	9.10	13.70	3.62	14.85
	SA Mean	% dif. C_B	26.43	34.57	20.46	14.13	28.69	36.01	44.97	3.06
		% dif. C_R	22.43	−3.73	20.81	15.32	5.91	9.07	−5.55	22.89
		% dif. C_O	23.44	3.77	20.72	15.00	11.07	15.95	5.80	16.34

Table 13 Mixed GA. Statistical summary of the overall costs obtained with the instances of 40, 80, 120 and 163 collection points. In each case, 30 independent runs have been performed. The average runtime is in hours

Instance	Min	Median	Mean	Confidence interval		Runtime (h)	S-W test
$i.40.1(\lambda = 500)$	534.9	544.6	545.3	543.0	547.65	0.62	0.6906
$i.80.1(\lambda = 1000)$	1099	1123	1123	1119	1127	0.96	0.1295
$i.120.1(\lambda = 5000)$	1720	1749	1749	1746	1762	2.81	0.0574
$i.163.1(\lambda = 10000)$	2414	3036	3139	2770	3360	3.87	0.0018

on the other hand, with the same number of fitness function evaluations, SA obtains better solutions than GA, both in terms of minima and means based on a sample of 30 independent runs.

The fact that exact models obtain better solutions than metaheuristics indicates that the implemented MILP and MIQP formulations are well suited to the problem under consideration. Metaheuristics are generally applied to problems that do not have a specific algorithm or heuristic that gives a satisfactory solution (which is not the case); or when, because of the number of variables involved, finding an optimal solution with these algorithms is extremely difficult or even impossible in a reasonable time. This is the case of large instances considered for the problem addressed in this paper.

Tables 13 and 14 summarize the statistical results obtained from a sample of 30 independent runs for the instances of 40, 80, 120 and 163 collection points, with GA and SA, respectively. Furthermore, since the value of the objective function is higher the larger the instance considered, different values of λ were considered to evaluate the fitness function (7) (see Table 13). Specifically, Table 13 shows, for each sample of the 30 best overall costs (US$) obtained with the 30 independent mixed GA runs: the minimum, median, mean, 95 percent confidence interval of the mean or of the (pseudo)median, as well as the mean runtime (in hours) and the p-value obtained in the Shapiro Wilk (S-W) test with the **shapiro.test()** function. Again, as in Table 8, based on this p-value, the 95 percent confidence interval is calculated for the (pseudo)median with the **wilcox.test()** function, if p-value < 0.05, or for the mean with the **t.test()** function, otherwise. Table 14 also shows the initial temperature, T_0, and the number of fitness function evaluations, FFE, for each instance. Precisely this number of fitness function evaluations was used as the stopping criterion for GA in order to compare the performance of both algorithms. It should be noted that feasible solutions were obtained in all runs and a linear growth of the average runtimes is observed as the number of collection points increases. The GA slope is around 0.028 and the SA slope is smaller, around 0.017.

The comparison of the best solution obtained with both metaheuristics and the corresponding mean values are shown in Table 15, where the percentage difference between the costs obtained with GA and SA has been calculated with Eq. (14). It can be concluded that, for the same number of evaluations of the fitness function, SA obtains better solutions than GA, both in terms of minimum and mean values of the overall and routing costs based on samples of 30 independent runs. Furthermore, the greatest percentage differences are achieved with the $i.163.1$ instance, these being 8.75% (minimum) and 9.26% (mean) for the overall costs, and 13.24% (minimum) and 13.82% (mean) for the routing costs.

Table 14 Mixed SA. Statistical summary of the overall costs obtained with the instances of 40, 80, 120 and 163 collection points. In each case, 30 independent runs have been performed. The average runtime is in hours

Instance	Min	Median	Mean	Confidence interval		Runtime (h)	S-W test	T_0	FFE
$i.40.1$	497.5	508.5	508.0	505.6	510.4	0.596	0.5182	1,960	1.020e06
$i.80.1$	1034	1055	1055	1050.9	1059.5	1.188	0.5634	12,547	1.105e06
$i.120.1$	1609	1638	1637	1631.5	1643.4	1.830	0.4604	17,044	1.120e06
$i.163.1$	2220	2264	2263	2254.4	2270.8	2.652	0.7309	28,871	1.145e06

Table 15 Large instances. SA versus GA

		Cost	$i.40.1$	$i.80.1$	$i.120.$	$i.163.1$
	GA Min	Bin (C_B)	179.29	336.64	507.48	673.87
		Routing (C_R)	355.57	762.11	1212.27	1740.37
		Overall (C_O)	534.86	1098.75	1719.75	2414.24
	GA Mean	Bin (C_B)	172.59	333.20	497.49	670.97
		Routing (C_R)	372.70	789.96	1256.50	1801.09
		Overall (C_O)	545.29	1123.16	1753.99	2472.06
	SA Min	Bin (C_B)	175.68	333.12	507.69	683.05
		Routing (C_R)	321.78	700.86	1101.11	1536.90
		Overall (C_O)	497.46	1033.98	1608.80	2219.95
	SA Mean	Bin (C_B)	173.21	335.65	505.86	680.23
		Routing (C_R)	334.83	719.52	1131.55	1582.36
		Overall (C_O)	508.04	1055.17	1637.41	2262.59
SA vs GA	Min	% dif. C_B	2.05	1.06	−0.04	−1.34
		% dif. C_R	10.50	8.74	10.10	13.24
		% dif. C_O	7.52	6.26	6.90	8.75
	Mean	% dif. C_B	−0.36	−0.73	−1.65	−1.36
		% dif. C_R	11.31	9.79	11.04	13.82
		% dif. C_O	7.33	6.44	7.12	9.26

7 Conclusion and future work

Effective MSW systems are essential for contemporary cities, playing a critical role in enhancing the livability and sustainability of smart city initiatives. However, careful planning and operation are paramount to achieve this goal. This paper addresses, in an integrated manner, two fundamental and interrelated problems in waste management: the sizing of collection points and the design of weekly collection route which has a periodic basis. These problems are often treated separately due to their inherent computational complexity.

In this work, two mathematical formulations for this problem are presented: a Mixed-Integer Quadratic model and a Mixed-Integer Linear model. In addition, a Genetic Algorithm and a Simulated Annealing algorithm, both with a complex chromosome representation, are proposed to handle the integrated problem. For the GA, extensive computational experiments were conducted, evaluating different crossover and mutation operators with various probabilities to determine the optimal configuration for the target problem. Similarly, for the SA, a design of experiments was carried out to evaluate the performance of different solution update methods and cooling factors. Both experimental designs were carried out with one of the smallest instances and the "winning configurations" were used with all the instances. Specifically, experiments were carried out with 5 instances of 12 collection points, involving 2,366 binary variables, 3 instances of 15 collection points, involving 3,584 binary variables and instances of 40, 80, 120 and 163 collection points, involving, respectively: 47,068; 367,416; 1,229,844 and 3,200,624 binary variables. All these scenarios are based on real data from the city of Bahía Blanca, Argentina.

Due to the number of binary variables involved, only a comparative analysis between the mathematical formulations and the metaheuristics was conducted with the small instances.

Between the exact models – with an execution time limit of 8 hours – the MILP formulation obtained solutions with better overall and routing costs in six of the eight instances. Furthermore, both exact models outperformed the SA and GA with all small instances and, with the same number of fitness function evaluations, SA obtains better solutions than GA, both in terms of minima and means based on a sample of 30 independent runs. The fact that exact models obtain better solutions than metaheuristics indicates that the implemented MILP and MIQP formulations are well suited to the problem under consideration. Metaheuristics are generally applied to problems that do not have a specific algorithm or heuristic that gives a satisfactory solution (it has been proven that this is not the case); or when, because of the number of variables involved, finding an optimal solution with these algorithms is extremely difficult or even impossible in a reasonable time. This is the case of large instances considered for the problem addressed in this paper.

For large instances, for which no feasible solutions were obtained with exact models, the proposed metaheuristics demonstrated efficient performance, offering high-quality feasible solutions in practical runtimes. Furthermore, a linear growth of the average runtimes is observed as the number of collection points increases. The GA slope is around 0.028 and the SA slope is smaller, around 0.017. This highlights their efficiency in handling problems in which exact models become computationally intractable or excessively time-consuming.

Taking into account that GA is the most widely used algorithm for solving the VRP and its variants, it is very interesting that, with the same number of evaluations of the fitness function, with the problem addressed in this paper, SA has obtained better solutions than GA with all the instances (small and large) and both in terms of minimum and mean values of the overall and routing costs, based on samples of 30 independent runs. Furthermore, the greatest percentage differences are achieved with the highest instance, these being 8.75% (minimum) and 9.26% (mean) for the overall costs, and 13.24% (minimum) and 13.82% (mean) for the routing costs.

Future research endeavors include enhancing the mathematical formulations and exact resolution with decomposition approaches, exploring other metaheuristics such as Tabu Search for comparative analysis, and potentially addressing the problem as a bi-objective optimization, to separately optimize bin installation and maintenance cost and routing cost. Furthermore, the problem of increased runtimes when using metaheuristics can be tackled from the point of view of parallelization of the algorithms. Another interesting line of research is solving the problem by applying Robust Design Optimization to handle uncertainties, for example, in the amount of waste deposited daily at collection points.

Appendix A An illustrative example of a feasible solution for the waste collection problem

This section shows an illustrative (feasible) solution of the i.12.1 to analyze how the decodification function of the metaheuristics works. Table 16 presents the chromosome of a feasible solution for this instance of twelve collection points and a planning horizon of one week in which the drivers' day of rest is Sunday. Firstly, the selection of the bin combination for a given collection point i is made taking into account the maximum daily amount of waste accumulated in the planning horizon $w_i^{max} = \max_{t \in T} w_{it}$. For example, $w_1^{max} = 5.08$, therefore, at collection point 1 it will be necessary to install a bin combination with a capacity of 5.08 m^3 or more, in this case, the one identified as 7 (see Table 2) . For establishing the routes performed each day t we follow the order of visitation of the collection points set by

Table 16 A feasible solution (chromosome) solution of instance $i.12.1$

Collection points	Bin combination	MON			TUE			WED		
		p_{i0}	m_{i0}	w_{i0}	p_{i1}	m_{i1}	w_{i1}	p_{i2}	m_{i2}	w_{i2}
1	7	8	0	2.54	0	0	3.81	4	1	5.08
2	7	6	1	3.24	4	1	1.62	0	0	1.62
3	2	5	1	2.34	3	1	1.17	2	1	1.17
4	6	9	0	2.98	2	1	4.47	5	0	1.49
5	6	10	0	3.18	5	1	4.77	6	0	1.59
6	5	2	1	2.42	8	0	1.21	7	0	2.42
7	7	1	1	5.28	9	0	1.32	8	0	2.64
8	7	11	0	3.69	6	1	4.92	9	0	1.23
9	4	7	1	3.16	10	0	1.58	3	1	3.16
10	2	4	1	2.34	11	0	1.17	1	1	2.34
11	5	0	0	2.00	1	1	3.00	10	0	1.00
12	4	3	1	2.66	7	1	1.33	11	0	1.33
Collection points	Bin combination	THUR			FRI			SAT		
		p_{i3}	m_{i3}	w_{i3}	p_{i4}	m_{i4}	w_{i4}	p_{i5}	m_{i5}	w_{i5}
1	7	5	0	1.27	7	0	2.54	4	1	3.81
2	7	6	0	3.24	3	1	4.86	3	1	1.62
3	2	7	0	1.17	2	1	2.34	2	1	1.17
4	6	8	0	2.98	1	1	4.47	8	1	1.49
5	6	9	0	3.18	5	1	4.77	9	1	1.59
6	5	3	1	3.63	8	0	1.21	6	1	2.42
7	7	2	1	3.96	9	0	1.32	11	0	2.64
8	7	10	0	2.46	6	1	3.69	0	0	1.23
9	4	11	0	1.58	4	1	3.16	10	1	1.58
10	2	1	1	1.17	10	0	1.17	1	1	2.34
11	5	0	0	2.00	11	0	3.00	7	1	4.00
12	4	4	1	2.66	0	0	1.33	5	1	2.66

the chromosome $[y_{it}]$ and taking into account that $m_{it} = 0$ indicates that the collection point i is not visited on day t, therefore, this point must be ignored in the order of visitation. For example, on Monday only collection points 7, 6, 12, 10, 3, 2 and 9 will be visited, in that order. The number of routes for these visits is established taking into account the capacity of the collection vehicles (12 m^3 in this case). A route ends when the vehicle does not have the capacity to empty the bin combination of the next collection point. Fig. 8 shows the routes associated with the solution shown in Table 16 and Table 17 shows the waste collected by the vehicle on each route of that solution.

Finally, the routes performed by the collection truck are presented in Figure 9. Routes are performed from Monday to Saturday (on Sundays no collection is performed). As was represented in Fig. 8, on Mondays, Thursdays, Fridays and Saturdays, two routes are performed. However, on Wednesday and Thursday only one route is performed. Due to the cycle feature of the proposed PCVRP model, this weekly schedule is repeated every week.

Table 17 Time and waste collected by the vehicle on each route of the solution shown in Table 16. Index g_i represents the previous collection point on the route (or the depot, if i is the first point on the route). The collection vehicle capacity is 12 m^3.

MON	R1	i	0	7	6	12	0			25.04 min
		w_{i0}		5.28	2.42	2.66				
		$v_{g_i i10}$		0.00	5.28	7.70	10.36			
	R2	i	0	10	3	2	9	0		23.05 min
		w_{i0}		2.34	2.34	3.24	3.16			
		$v_{g_i i20}$		0.00	2.34	4.68	7.92	11.08		
TUE	R1	i	0	11	4	3	2	0		26.00 min
		w_{i1}		3.00	4.47	1.17	1.62			
		$v_{g_i i11}$		0.00	3.00	7.47	8.64	10.26		
	R2	i	0	5	8	12	0			22.29 min
		w_{i1}		4.77	4.92	1.33				
		$v_{g_i i21}$		0.00	4.77	9.69	11.02			
WED	R1	i	0	10	3	9	1	0		25.80 min
		w_{i2}		2.34	1.17	3.16	5.08			
		$v_{g_i i12}$		0.00	2.34	3.51	6.67	11.75		
THUR	R1	i	0	10	7	6	12	0		26.00 min
		w_{i3}		1.17	3.96	3.63	2.66			
		$v_{g_i i13}$		0.00	1.17	5.13	8.76	11.42		
FRI	R1	i	0	4	3	2	0			23.41 min
		w_{i4}		4.47	2.34	4.86				
		$v_{g_i i14}$		0.00	4.47	6.81	11.67			
	R2	i	0	9	5	8	0			22.67 min
		w_{i4}		3.16	4.77	3.69				
		$v_{g_i i24}$		0.00	3.16	7.93	11.62			
SAT	R1	i	0	10	3	2	1	12	0	24.26 min
		w_{i5}		2.34	1.17	1.62	3.81	2.66		
		$v_{g_i i15}$		0.00	2.34	3.51	5.13	8.94	11.60	
	R2	i	0	6	11	4	5	9	0	29.99 min
		w_{i5}		2.42	4.00	1.49	1.59	1.58		
		$v_{g_i i25}$		0.00	2.42	6.42	7.91	9.50	11.08	

Waste Collection Routes															
MON	R1	0	→	7	→	6	→	12	→	0	(25.04 min)				
MON	R2	0	→	10	→	3	→	2	→	9	→	0	(23.05 min)		
TUE	R1	0	→	11	→	4	→	3	→	2	→	0	(26.00 min)		
TUE	R2	0	→	5	→	8	→	12	→	0	(22.29 min)				
WED	R1	0	→	10	→	3	→	9	→	1	→	0	(25.80 min)		
THUR	R1	0	→	10	→	7	→	6	→	12	→	0	(26.00 min)		
FRI	R1	0	→	4	→	3	→	2	→	0	(23.41 min)				
FRI	R2	0	→	9	→	5	→	8	→	0	(22.67 min)				
SAT	R1	0	→	10	→	3	→	2	→	1	→	12	→	0	(24.26 min)
SAT	R2	0	→	6	→	11	→	4	→	5	→	9	→	0	(29.99 min)

Fig. 8 Waste collection routes associated with the solution shown in Table 16. The time taken to complete them (in minutes) is also shown

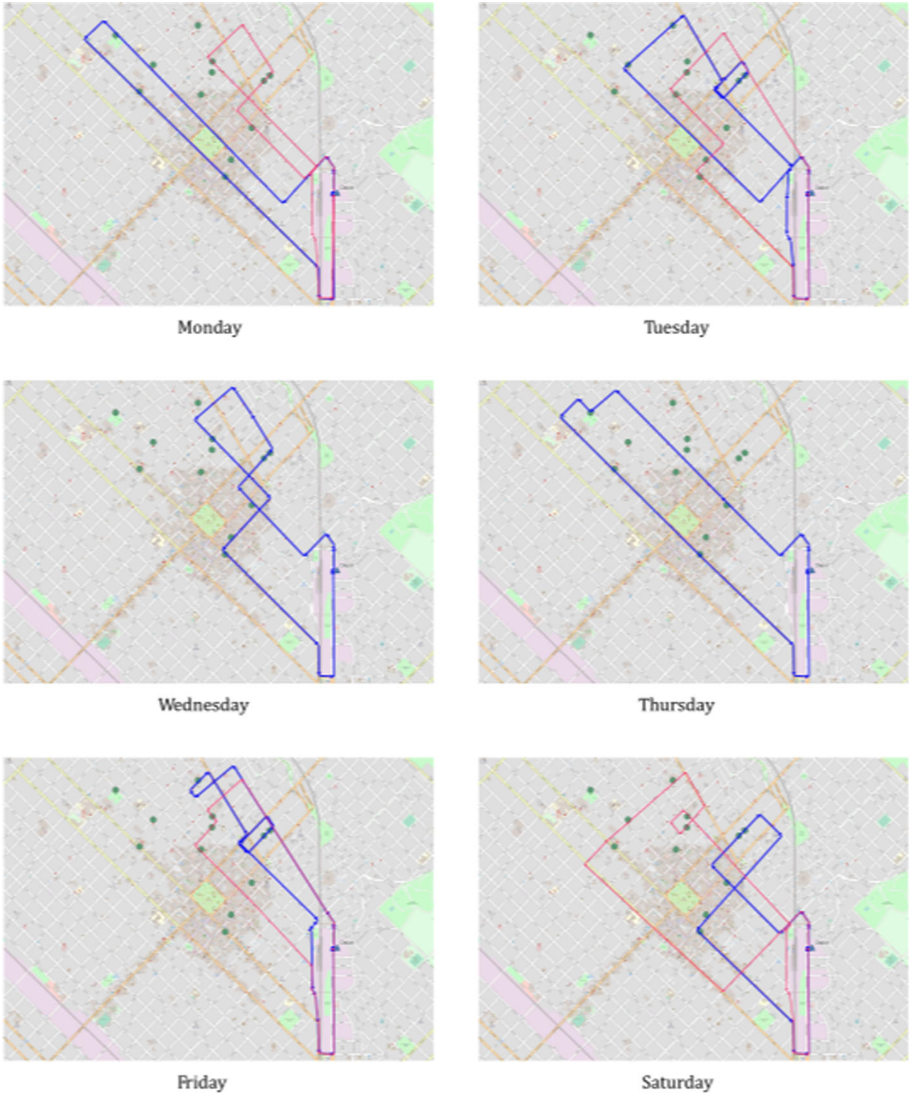

Fig. 9 Routes performed by the collection truck each day of the time horizon

Author Contributions Conceptualization: G.B., R.D., F.M. and M.M.; Methodology: G.B. and R.D.; Formal analysis and investigation: G.B. and R.D.; Writing - original draft preparation: G.B. and R.D.; Writing - review and editing: G.B., R.D., F.M. and M.M.; Funding acquisition: G.B. and M.M.; Resources: G.B., R.D., F.M. and M.M.; Supervision: G.B., R.D., F.M. and M.M.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The authors are grateful for the financial support granted by the Consejería de Economía, Industria, Comercio y Conocimiento of the Government of the Canary Islands through the direct grant awarded to the ULPGC called "Support for R+D+i activity. Campus of International Excellence CEI CANARIAS-ULPGC".

Data Availability This paper presents results obtained with several instances based on data collected in field studies in the city of Bahía Blanca, Argentina (Cavallin et al., 2020), and they can be downloaded from https://github.com/diegorossit/ANOR-S-24-01950.git.

Declarations

Conflicts of Interest The authors declare that they have no conflicts of interest.

Consent for publication All authors have read and agreed to the published version of the manuscript.

References

Archetti, C., & Ljubić, I. (2022). Comparison of formulations for the inventory routing problem. *European Journal of Operational Research, 303*(3), 997–1008. https://doi.org/10.1016/j.ejor.2021.12.051

Asociación Latinoamericana de Logística (sept-2023). Indicadores Costos Logísticos países miembros de ALALOG. Available in https://www.alalog.org/es/studies

Beltrami, E., & Bodin, L. (1974). Networks and vehicle routing for municipal waste collection. *Networks, 4*, 65–94.

Brogaard, L., & Christensen, T. (2012). Quantifying capital goods for collection and transport of waste. *Waste management & research, 30*(12), 1243–1250. https://doi.org/10.1177/0734242X12462279

Bynum, M., Hackebeil, G., Hart, W., Laird, C., Nicholson, B., Siirola, J., Watson, J., & Woodruff, D. (2021). Pyomo-optimization Modeling in Python vol. 67. Springer

Carlos, M., Gallardo, A., Edo, N., & Abaso, J. (2019). Influence of the municipal solid waste collection system on the time spent at a collection point: A case study. *Sustainability, 11*(22), 6481. https://doi.org/10.3390/su11226481

Cavallin, A., Rossit, D., Herran, V., Rossit, D., & Frutos, M. (2020). Application of a methodology to design a municipal waste pre-collection network in real scenarios. *Waste Management & Research, 38*(1-suppl), 117–129. https://doi.org/10.1177/0734242X19894630

Cornuéjols, G., Sridharan, R., & Thizy, J. (1991). A comparison of heuristics and relaxations for the capacitated plant location problem. *European journal of operational research, 50*(3), 280–297.

Cubillos, M., & Wøhlk, S. (2020). Solution of the maximal covering tour problem for locating recycling drop-off stations. *Journal of the Operational Research Society, 72*(8), 1898–1913. https://doi.org/10.1080/01605682.2020.1746701

Davis, L. (1985). Applying adaptive algorithms to epistatic domains. In: International Joint Conferences on Artificial Intelligence (IJCAI), pp. 162–164.

Delahaye, D., Chaimatanan, S., & Mongeau, M. (2019). Simulated annealing: From basics to applications. In: Gendreau, M., Potvin, J.-Y. (eds.) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 272, pp. 1–35. Springer, Cham. https://doi.org/10.1007/978-3-319-91086-4_1

D'Onza, G., Greco, G., & Allegrini, M. (2016). Full cost accounting in the analysis of separated waste collection efficiency: A methodological proposal. *Journal of environmental management, 167*, 59–65. https://doi.org/10.1016/j.jenvman.2015.09.002

Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics, 50*(3), 239–254. https://doi.org/10.1016/0166-218X(92)00172-I

Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers & Industrial Engineering, 140*, Article 106242. https://doi.org/10.1016/j.cie.2019.106242

Goldberg, D., & Lingle, R. (2014). Alleles, Loci and the traveling salesman problem. In: First International Conference on Genetic Algorithms and Their Applications, pp. 154–159. Psychology Press

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management science, 22*(4), 455–460.

Gurobi Optimization, L. (2023). Gurobi Optimizer Reference Manual. https://www.gurobi.com

Gläser, S., & Stücken, M. (2021). Introduction of an underground waste container system-model and solution approaches. *European Journal of Operational Research, 295*(2), 675–689.

Hemmelmayr, V., Doerner, K., Hartl, R., & Vigo, D. (2014). Models and algorithms for the integrated planning of bin allocation and vehicle routing in solid waste management. *Transportation Science, 48*(1), 103–120. https://doi.org/10.1287/trsc.2013.0459

Hussain, A., Muhammad, Y., Nauman, M., Hussain, I., Mohamd, A., & Gani, S. (2017). Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Computational Intelligence and Neuroscience, 1–7. https://doi.org/10.1155/2017/7430125

Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. Cambridge: MIT Press.

Hemmelmayr, V., Smilowitz, K., & Torre, L. (2017). A periodic location routing problem for collaborative recycling. *IISE Transactions, 49*(4), 414–428. https://doi.org/10.1080/24725854.2016.1267882

Hollander, M., & Wolfe, D. (1973). *Nonparametric Statistical Methods*. New Jersey: John Wiley & Sons.

Han, J., Zhang, J., Guo, H., & Zhang, N. (2024). Optimizing location-routing and demand allocation in the household waste collection system using a branch-and-price algorithm. *European Journal of Operational Research, 316*(3), 958–975.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Maalouf, A., & Agamuthu, P. (2023). Waste management evolution in the last five decades in developing countries-A review. *Waste Management & Research, 41*(9), 1420–1434. https://doi.org/10.1177/0734242X231160099

Mahéo, A., Rossit, D., & Kilby, P. (2020). A Benders decomposition approach for an integrated bin allocation and vehicle routing problem in municipal waste management. Communications in Computer and Information Science, 3–18. https://doi.org/10.1007/978-3-030-76310-7_1

Mahéo, A., Rossit, D., & Kilby, P. (2023). Solving the integrated bin allocation and collection routing problem for municipal solid waste: A Benders decomposition approach. *Annals of Operations Research, 322*(1), 441–465. https://doi.org/10.1007/s10479-022-04918-7

Metropolis, N., Rosenbluth, A., Rosenbluth, M. N., Teller, A., & Teller, E. (1953). Equation of state calculation by fast computing machines. *J. Chem. Phys., 21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Niu, Y., Xu, C., Liao, S., Zhang, S., & Xiao, J. (2024). Multi-objective location-routing optimization based on machine learning for green municipal waste management. *Waste Management, 181*, 157–167. https://doi.org/10.1016/j.wasman.2024.04.001

Osman, I., & Kelly, J. (1996). *Meta-Heuristics: Theory and Applications*. Boston/London/Dordrecht: Kluwer Academic Publishers.

Owusu, F., Oduro, S., Essandoh, H., Wayo, F., & Shamudeen, M. (2019). Characteristics and management of landfill solid waste in Kumasi, Ghana. Scientific African **3**(e00052). https://doi.org/10.1016/j.sciaf.2019.e00052

Oliver, I., Smith, D., & Holland, J. (1987). A study of permutation crossover operators on the traveling salesman problem. In: Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application, pp. 224–230. Psychology Press

Posit team (2023). RStudio: Integrated Development Environment For R. Posit Software, PBC, Boston, MA. Posit Software, PBC. http://www.posit.co/

R Core Team (2022). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. R Foundation for Statistical Computing. https://www.R-project.org/

Rossit, D., González Landín, B., Frutos, M., & Méndez Babey, M. (2024). An allocation-routing problem in waste management planning: Exact and heuristic resolution approaches. In: Nesmachnow, S., Hernández Callejo, L. (eds.) Smart Cities. ICSC-CITIES 2023. Communications in Computer and Information Science, vol. 1938, pp. 92–107. Springer, Cham. https://doi.org/10.1007/978-3-031-52517-9_7

Roy, A., Manna, A., Kim, J., & Moon, I. (2022). IoT-based smart bin allocation and vehicle routing in solid waste management: A case study in South Korea. *Computers & Industrial Engineering, 171*, Article 108457. https://doi.org/10.1016/j.cie.2022.108457

Rossit, D., & Nesmachnow, S. (2022). Waste bins location problem: A review of recent advances in the storage stage of the municipal solid waste reverse logistic chain. *Journal of Cleaner Production, 342*, Article 130793. https://doi.org/10.1016/j.jclepro.2022.130793

Rossit, D., Nesmachnow, S., & Cavallin, A. (2023). Municipal solid waste management systems: Application of SWOT methodology to analyze an Argentinean case study. In: VI Ibero-American Congress of Smart Cities ICSC-CITIES 2023, Mexico City, pp. 668–680

Rossi, M., Papetti, A., & Germani, M. (2022). A comparison of different waste collection methods: Environmental impacts and occupational risks. *Journal of Cleaner Production, 368*, Article 133145. https://doi.org/10.1016/j.jclepro.2022.133145

Rossit, D., Tohmé, F., Frutos, M., & Broz, D. (2017). An application of the augmented ε-constraint method to design a municipal sorted waste collection system. *Decision Science Letters, 6*(4), 323–336. https://doi.org/10.5267/j.dsl.2017.3.001

Rodriguez, M., & Vecchietti, A. (2013). A comparative assessment of linearization methods for bilinear models. *Computers & Chemical Engineering, 48*, 218–233. https://doi.org/10.1016/j.compchemeng.2012.09.011

Siegel, S., & Castellan, N. (1988). *Non Parametric Statistics for the Behavioural Sciences*. New York: MacGraw Hill.

Toth, P., & Vigo, D. (2014). Vehicle Routing: Problems, Methods, and Applications. Society for Industrial and Applied Mathematics, Philadelphia. https://doi.org/10.1137/1.9781611973594