Expert Systems With Applications 145 (2020) 113101

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Iterated greedy with variable neighborhood search for a multiobjective waste collection problem

Laura Delgado-Antequera[a], Rafael Caballero[a], Jesús Sánchez-Oro[b], J Manuel Colmenar[b], Rafael Martí[c,*]

[a] Department of Applied Economics (Mathematics), Universidad de Málaga, Spain
[b] Department of Computer Science, Universidad Rey Juan Carlos, Spain
[c] Department of Statistics and Operations Research, Universidad de Valencia, Spain

ARTICLE INFO

Article history:
Received 5 April 2019
Revised 23 November 2019
Accepted 24 November 2019
Available online 26 November 2019

Keywords:
Metaheuristics
Multiobjective optimization
Vehicle routing

ABSTRACT

In the last few years, the application of decision making to logistic problems has become crucial for public and private organizations. Efficient decisions clearly contribute to improve operational aspects such as cost reduction or service improvement. The particular case of waste collection service considered in this paper involves a set of economic, labor and environmental issues that translate into difficult operational problems. They pose a challenge to nowadays optimization technologies since they have multiple constraints and multiple objectives that may be in conflict. We therefore need to resort to multiobjective approaches to model and solve this problem, providing efficient solutions in short computational times. In particular, we consider four different objectives to model the waste collection problem: travel cost, route length balance, route time balance, and number of routes. We propose an iterated greedy algorithm coupled with a variable neighborhood search to minimize an achievement function to determine a good approximation to the Pareto front. The performance of our method is empirically analyzed on a set of instances (both generated and real), and compared with the well-known NSGA-II and SPEA2 methods. The comparison favors our proposal.

1. Introduction

It is nowadays well-known that the growing population is highly correlated to the large amount of waste generated. This fact has encouraged public decision makers to put considerable effort in the study of Solid Waste Management (SWM), as the increasing generation of waste in modern economy is closely linked to the growth of production and consumption.

This research was motivated by a real problem in Antequera, a rural area of Spain. According to the Spanish regulation, waste generated in these areas should be collected by municipal administrations, which happen to have relatively limited budgets. This fact has encouraged the interest on waste collection and transportation optimization to achieve economic savings and improve the efficiency of the service. Mainly dedicated to the agriculture, the region where our problem is located hosts a population over 126,000 citizens, split into 24 different municipalities. The Waste Collection Problem (WCP) can be modeled with a set of collection points located through this area where citizens leave their garbage in containers (or bins) with some pre-specific dimension (determined in terms of the pick-up vehicles and the population size). In rural areas, containers are usually placed at the center of the municipality, in order to allow the corresponding vehicle to perform the service efficiently. Each collection point has one or more refuse bins. Different trucks collect each type of waste and transport the waste to their final destination (the landfill). Fig. 1 shows the distribution of 991 rear containers throughout this region, whose service is currently provided by a fleet of 3 rear trucks, which perform a set of 7 different routes departing from a unique depot.

Route planning for waste collection is currently done over the area of Antequera by experts. In particular, engineers with years of experience in waste collection are responsible for deciding all the parameters and constraints involved in the problem: number of trucks, route design, balance among routes, etc. According to their experience, it is a rather complicated problem with a relatively large number of constraints and objectives, which result in human mistakes. In this context, our goal is to offer them an efficient routing system according to the preferences, variables and constraints

* Corresponding author.
E-mail addresses: lauda1g10@uma.es (L. Delgado-Antequera), r_caballero@uma.es (R. Caballero), jesus.sanchezoro@urjc.es (J. Sánchez-Oro), josemanuel.colmenar@urjc.es (J.M. Colmenar), rafael.marti@uv.es (R. Martí).

https://doi.org/10.1016/j.eswa.2019.113101
0957-4174/

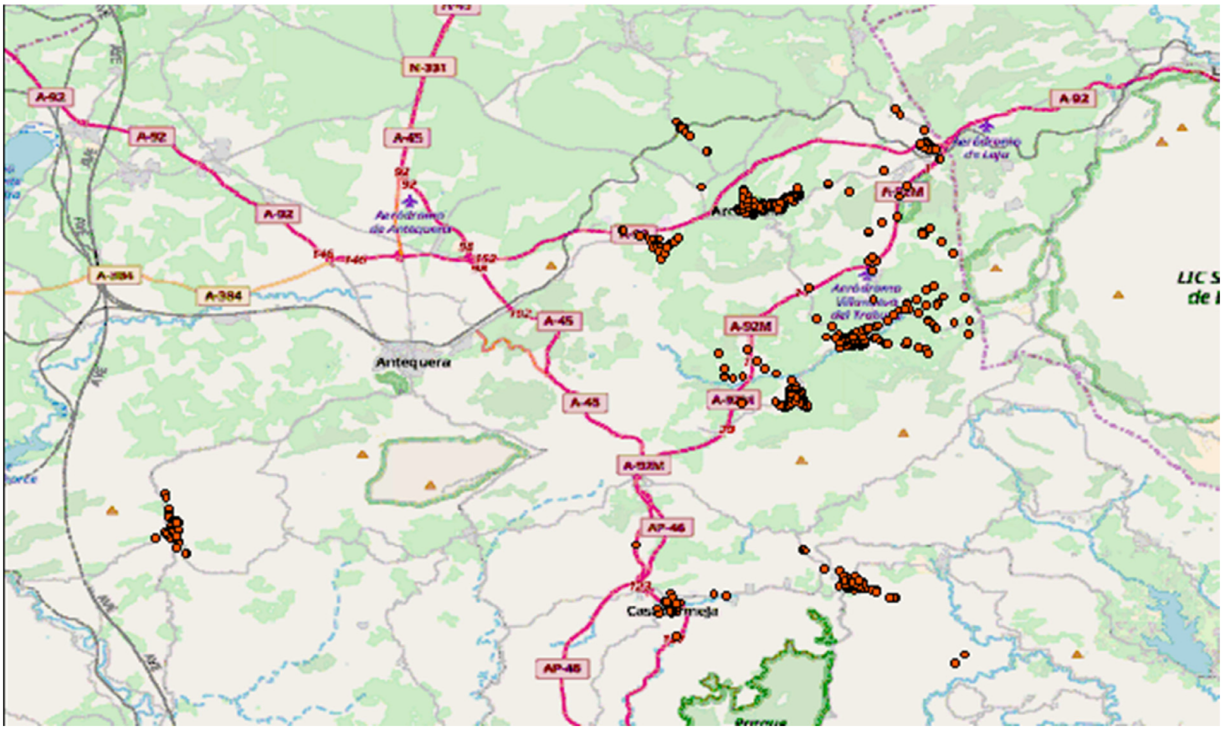

Fig. 1. Bin distribution in our real instance.

specified by the manager of the service in the region of Antequera. This involves considering multiple objectives to deal with different aspects related to social, environmental and economic factors.

Guerrero, Mass and Hogland (2013) present an analysis of waste management problems, taking into account different indicators, including the critical impact of waste management on the environmental climate. Moreover, economically speaking, waste collection and its transportation constitutes a large fraction of the total cost for the municipal administrations. Hence, multiple benefits would be obtained if they use a good decision making procedure that permits to improve the waste collection service. Additionally, there are real constraints given by road characteristics such as the speed, directions or turns, which have to be also considered in the resolution process.

As described in the review by Jozefowiez, Semet and Talbi (2007), a routing problem can be defined in terms of the following components: network, demand, fleet, cost, and objective. The network can be symmetrical or asymmetrical, indicating whether there are different links connecting the nodes. The demands can be fixed or stochastic and associated with nodes or arcs. The fleet can be heterogeneous or homogeneous and it generates constraints that affect the tours. The costs are generally variable depending on the vehicle use. The objective can be single or multiple, and in the latter case, the authors point out that since objectives frequently conflict, a multiobjective approach is advantageous. In this context, we find that many real-life optimization problems can hardly be considered as properly formulated without taking into account their multiple objective nature (Stewart et al., 2008). This is especially true in the case of vehicle routing problems, which in general terms are defined as generating tours on a network given a set of constraints while optimizing one or several objectives, which frequently conflict. Literature on vehicle routing (Jozefowiez et al., 2007) reveals a large number of publications regarding multiple criteria in different areas, such as humanitarian logistics, public transportation, waste collection or scholar transportation, among others. In this paper, we target the waste collection problem, in which to improve customer service, different criteria have been added to the classic travel distance.

It is clear that multiobjective problems are difficult to solve since most of the times they optimize several objectives which, individually considered, would result on NP-hard problems. This is the case of routing problems, which have been extensively studied in their many mono-objective versions. Nowadays, with the development of the metaheuristic technologies, more realistic problems with several objectives have been approached. As a result, we can offer practical solutions to the industries and government, thus contributing to the efficiency of the so-called supply chain or providing better services to the citizens.

In a multiobjective problem (MOP), a set of $m \geq 2$ functions $\{f_1(x),\ldots, f_m(x)\}$ are simultaneously optimized (without loss of generality we can consider them to be minimized). This problem can be defined in mathematical terms as:

$$minimize\ \{f_1(x), \ldots, f_m(x)\}$$

$$s.t:$$

$$x \in S$$

where $S \subseteq R^n$ is the set of feasible solutions.

It is well documented that we cannot expect to find a solution that optimizes all the objectives in a MOP at the same time given that the objectives are usually in conflict. Pareto optimality principle describes an equilibrium for a solution where the value of no objective can be improved without deteriorating the value of another objective. Hence, to compare two solutions $z^1, z^2 \in R^m$ in the objective space, we say that z^1 *dominates* z^2 if $z_i^1 \leq z_i^2\ \forall i \in \{1, 2\ldots m\}$ and $\exists j \in \{1, 2\ldots m\}$ such that $z_j^1 < z_j^2$. Otherwise, we say that z^1, z^2 are (mutually) *non-dominated*. According to this principle, a feasible solution x^1 is *Pareto optimal* (also called *efficient*) if there is no other solution x^2 such that $z^2 = f(x^2)$ dominates $z^1 = f(x^1)$. The set of all the Pareto optimal solutions, E, is called *Pareto optimal set* and its image in the objective space, $f(E)$, is referred to as the *Pareto optimal front*.

There are several ways to approach a multiobjective problem. The weighted sum of the objectives is probably the most intuitive although it is well known that it provides poorly distributed solutions and does not find Pareto optimal solutions in non-convex regions. Another classic method is the ε-constraint where we only optimize one objective and the others are modeled as constraints with certain threshold values (which can be difficult to set in the case of many objectives). More recently, Pareto methods directly use the notion of dominance, by applying metaheuristic algorithms to identify a large set of non-dominated points. Evolutionary approaches (Stewart et al., 2008), such as genetic algorithms, have been widely applied in this context (García-Najera & Bullinaria, 2011), but we can also find other methods such as tabu search (Pacheco & Martí, 2006), scatter search (Caballero, Laguna, Martí & Molina, 2011), GRASP (Martí, Campos, Resende & Duarte, 2015), or ant colony optimization (Bautista, Fernández & Pereira, 2008) with remarkable results.

In this paper, we approach multiobjective optimization with a solution procedure based on the metaheuristic methodology called Iterated Greedy (IG) (Ruiz & Stützle, 2007) coupled with Variable Neighborhood Search (VNS) (Mladenovic & Hansen, 1997). Recent studies have demonstrated their practical advantages for solving a diverse array of optimization problems from both classical and real world settings, including problems in routing and scheduling. For instance, Liao and Lin (2003) present an implementation for makespan minimization in distributed flowshops, Ying and Cheng (2010) addresses the scheduling of dynamic parallel machines whose setup times may vary, and García-Martinez, Glover, Rodriguez, Lozano and Martí (2014) considered the multiple knapsack problem. On the other hand, Variable Neighborhood Search has been considered for several heterogeneous problems, from location of obnoxious facilities (Herrán, Colmenar, Martí & Duarte, 2020), to graph drawing (Sánchez-Oro, Martínez-Gavara, Laguna, Martí & Duarte, 2017).

Our metaheuristic solving method relies on the Wierzbicki achievement scalarizing function (ASF) to guide the search (Wierzbicki, 1980). As we will discuss in Section 4, a large variation of the weights assigned to the objectives permits to perform an efficient search of the multiobjective solution space. The combination of metaheuristics with ASF constitutes our main contribution and, as shown in the computational experiments, it is able to obtain high quality approximations of the Pareto set in short computational times

In the following section, we provide a short background on multiobjective routing problems. In Section 3 we describe our proposal to implement the Wierzbicki's ASF. Section 4 is devoted to the methodology, and Section 5 to the solving method based on IG and VNS. Section 6 evaluates the performance of our proposal by comparing it with other multiobjective methods on a large set of instances. In particular, a deep and extensive comparison with the NSGA-II and SPEA2 algorithms on public domain instances is performed. Finally, Section 7 shows the results obtained for the real instance, using the best algorithm discussed in the previous section.

2. Literature review

The scientific study of Waste Management can be traced back to the seventies. Marks and Liebman (1970) highlighted some research lines that could be addressed in the field of Operational Research. This fact aware the sanitary departments of big cities like New York or Washington D.C., which started several studies focused on developing operational research strategies to improve their services. In particular, routing problems constitute an important and huge family of combinatorial optimization problems, that can be applied to solve waste collection problems (WCP), which roughly speaking contain both city and rural instances. In the former case, due to the distribution of the bins almost continuously along the streets, WCP are usually modeled as *Capacitated Arc Routing Problems (CARP)*. For further information about the application of CARP to solve WCP, see: Constantino, Gouveia, Ao and Nunes (2015), Corberan and Laporte (2015) or Cortinhal, Mourao and Nunes (2016), Hanafi, Freville and Vaca (1999), Marks and Liebman (1970), among others. However, in rural instances, as it is our case in this paper, containers are in specific locations relatively far apart from each other, and therefore they are modeled as *Capacitated Vehicle Routing Problems (CVRP)*. In a recent survey, Cattaruzza, Nabil, Dominique and González-Feliu (2017) present a complete taxonomy on urban goods movement where waste collection is classified into urban management movements. These authors state that the flows of waste collection are highly dependent on the waste nature, and the optimization problem is usually defined depending on the type of waste. In particular, the WCP belongs to the garbage collection, which is suggested to be modeled as a VRP.

In the past, solid waste collection did not use to take into account an analysis of demand, and the construction of the routes was left to the drivers. Cities, however, continue to expand and the importance of an efficient collection system increases accordingly. Then, in spite that it is hard to provide the perfect solution, different methods have been developed that focus on route length, costs, number of collection vehicles, and other criteria. In general, there is a natural subdivision when studying the WCP: (i) Determination of the frequency to visit each location and (ii) definition of the optimum set of routes to service all the corresponding locations every day.

Angelelli and Speranza (2002) propose a model that incorporates the frequency and considers some hygiene requirements. Their method is based on a Tabu Search with neighborhoods defined by the shift operator. Also, Baptista et al. (2002) developed a heuristic to maximize the benefits obtained from a periodical recycling paper collection. An interesting study (Maniezzo & Roffilli, 2008) defines a methodology to transform CARP into CVRP. Their aim is to minimize the overall distance travelled by a fleet of trucks, subject to time windows and allowing multiple trips. The authors use a multi-start heuristic combined with Variable Neighborhood Search to solve the resulting CVRP. The multi-start algorithm constructs an initial solution from scratch and improves it with Tabu Search.

Complex waste management systems in general, and in particular sitting waste management and disposal facilities has recently been a preferential field of Geographical Information Systems (GIS) applications (Marshall & Farahbakhsh, 2013; Bing et al., 2016). We can find different metaheuristics implemented within GIS packages, such as: Sharma insertion algorithm (Ghose, Dikshit & Sharma, 2006) to allocate three different types of bins and vehicles for the Asansol Municipality Corporation (AMC) of West Bengal State (India); or the Ant Colony System to optimize routes and provide the most cost - effective itinerary (Karadimas, Papatzelou & Loumos, 2007; Li & He, 2009). In this context, Tavares, Zsigraiova, Semiao and Carvalho (2009) incorporate fuel consumption to the use of 3D route modelling within ArcGIS Network Analysis (NA). Recently, Nguyen-Trong, Nguyen-Thi-Ngoc, Nguyen-Ngoc and Dinh-Thi-Hai (2017) have provided a solution to model Vietnam waste collection system with a successful cost reduction and Erfani, Danesh, Karrabi and Shad (2017) have proposed another model in order to improve bins distribution and vehicle routing for the Municipal Solid Waste Problem.

There are many applications of multicriteria algorithms in the design of routes of real problems. For instance, Samanlioglu (2013) applies a lexicographic Tchebycheff formulation, using CPLEX, in order to optimize the total transportation

Table 1
Multiobjective waste collection literature review.

References	Objectives
Chang and Wei (1999)	min number and size of drop-off stations; maximize the population covered with the service; maximize the average walking distance from the population to drop - off stations; min travel distance
Garcia-Najera and Bullinaria (2011), Tung and Pinnoi (2000)	min total cost; min number of vehicles
Kim et al. (2006)	min number of vehicles; min total travelling time; max route compactness; max workload balance
Ombuki, Ross and Hanshar (2006)	min total cost; min number of vehicles
Alumur and Kara (2007)	min total cost; min transportation risk
Chalkias and Lasaridi (2009)	min time; min distance travelled; min man – effort
Arribas et al. (2010)	min labor; min operation cost; min transport costs
Hemmelmayr et al. (2013, 2014)	trade - off between the service frequency over a planning period and the number of bins that can be placed there
Li, Leung, Lin and Huang (2013a)	min travel time; min accident probability; min off road population and special needs at risk; min expected economic damage; road users at risk
Li et al. (2013b)	min travel time; min accident probability; min off road population exposure and special needs at risk; min expected economic damage; road users at risk
Ferreira et al. (2015)	min total distance; max amount of waste; max amount of waste collected / km; max number of eco-points visited; min number of vehicles; max number of priority points collected
Xue and Cao (2016)	min total cost; min accident cost; min accident risk; min exposure to public
Gómez, Pacheco and Gonzalo-Orden (2015)	min total cost; min amount of waste accumulated in a period of time
López-Sánchez et al. (2017)	min *total cost; route balance*

cost of hazardous materials and waste residues, the total transportation risk and site risk. In particular, if we focus on the transportation of solid waste with multiple criteria and despite the wide information provided in the literature reviews: Beliën, De Boeck and Van Ackere (2012), we highlight several papers that obtain the Pareto optimal front for multi-objective waste collection problems. Table 1 summarizes the most relevant works related to waste collection and highlights the objectives considered in each one.

As in other multiobjective problems, it is common to find applications of genetic algorithms or other evolutionary methods in combination with different multicriteria strategies. This is the case of (Garcia-Najera & Bullinaria, 2011), where time windows are considered in a model optimizing total (sum) distance, total travel time and number of vehicles, or Ombuki, Ross and Hanshar (2006) that optimize the total distance and the number of vehicles by introducing a genetic algorithm based on Beasley's approach. Another metaheuristic, that belongs to the family of population based metaheuristics, has recently been developed in Xue & Cao (2016) where using GIS tools, the authors define a multiobjective Ant Colony Optimization method coupled with a min - max model and the Djistra's algorithm. We have also found other metaheuristic methodologies applied to these difficult problems, such as GRASP (López-Sánchez, Hernández-Díaz & Gortázar, 2017), VNS, or Tabu Search, as shown below.

Kim, Kim and Sahoo (2006) provide a realistic model, which includes lunch break, a single depot and multiple trips, while optimizing the number of vehicles, total travelling time, route compactness and workload balance to solve a commercial WCP in North America. The workload balance is measured as the difference between the longest and shortest route in terms of traveling times. After its publication, other authors considered this problem as a benchmark to test the performance of their methodologies. In this context, we find the works of Benjamin and Beasley (2010) and Benjamin and Beasley (2013), proposing a Variable Neighborhood Search (VNS) and Tabu Search (TS) respectively. These methods improve the computational time by pre-evaluating facilities insertions in a disposal facility positioning procedure and reduce the search space by determining a number of nearest - in time - nodes for each unrouted customer.

To solve a separate WCP in Austria, Hemmelmayr, Doerner, Hartl and Rath (2013) define a VNS algorithm, with the aim of estimating frequencies and considering intermediate facilities. Later, Hemmelmayr, Doerner, Hartl and Vigo (2014) incorporates dynamic programming to the process, in order to balance the tradeoff between the service frequency over a planning period and the number of bins that can be placed there. Other aspects are contemplated in this work such as multiple waste type, the number of bins allocated on a specific area, the capacity or volume of bins or the cost associated to a service.

Another interesting approach minimizes labor, operation and transportation costs (Arribas, Andrea Blazquez & Alejandra Lamas, 2010). The resolution process is carried out in three phases, so that it first uses the regret function and local search to construct clusters of the set of containers to be collected. Note that, in this construction, each cluster is obtained attending to the vehicle maximum capacity and the service schedule. Then, VRP is solved for each cluster using Tabu Search and, finally, an exact Branch and Bound method is applied to minimize the number of vehicles, considering that each cluster should be served by only one vehicle.

Faccio, Persona and Zanin (2011) propose a framework about the traceability technology available in the optimization of solid waste collection, as well as a VRP model with the real time traceability data of an Italian city. Das and Bhattacharyya (2015) propose a complete real waste transportation model for Kolkata metropolitan city. They divide the problem into different levels of transportation where each of them aims to minimize cost using local search strategies.

As expected, also many Decision Support Systems (Ferreira, Costa, Tereso & Oliveira, 2015; Simonetto & Borenstein, 2007; among others) have been designed, in combination with Geographical Information Systems, as a tool to share the information obtained with the decision maker (DM), and guide the decision process to the most preferred solution. They mainly focus on reducing the total distance traveled. Most of these works face the location of disposal facilities, as well as the design of an optimum route system. One key factor for proper operation of collection systems is the route design for each type of garbage, which may or may not be separately collected Bautista et al. (2008). We refer the reader to the recent survey by Han and Ponce-Cueto (2015).

3. Problem formulation

Waste collection is mainly a Vehicle Routing Problem (VRP), so any solution has to perform the service for a number of customers with a fleet of vehicles (Beliën et al., 2012). A vehicle routing problem typically consists of a set of vehicles, a pre-established set of stops and a depot. In a solution, each vehicle starts from the depot, visits a number of stops and ends at the depot. Depending on the complexity of the problem one can add different characteristics such as different types of vehicles, number of disposal facilities (single or multiple), various types of constraints, etc. Then, different models and methods can be applied to solve routing problems, such as integer linear programming, solution methods based on well-known routing models, such as the travelling salesman or the Chinese postman problems, hierarchical methods, and a wide collection of heuristics and metaheuristics.

From the analysis of the objectives covered in Table 1, and the discussion with the local administration corresponding to the rural instance that triggered our interest, in order to provide a wide overview of the different options available for the current WCP in Antequera, saving costs and introducing improvements on employees' conditions, as well as the quality of the service, we propose a model based on the following four objectives:

- f_1: To minimize total distance, computed as the sum of the overall route distances.
- f_2: To balance routes in terms of distance by the minimization of the longest route.
- f_3: To balance routes in terms of time by minimizing the difference between the duration of the longest and shortest routes.
- f_4: To minimize the number of routes.

Moreover, it must be noted that in the computation of f_3, we consider the stopping time to collect the waste in the picking points. We have empirically found that the inclusion of this time results in routes with similar number of stops, which is not observed when only optimizing f_2, in which this time is not considered. Finally, f_3 is necessary since in the WCP all workers must have equivalent working hours.

Currently, the waste collection in Antequera does not provide a daily service. However, the growing population within this area has forced the administrations to study the cost associated to anticipate the situation in which we have to empty each container every day. Then, some of the main features of the real problem can be properly formulated in mathematical terms using the following notation:

- The set of nodes to visit $N=\{1, 2, \dots n\}$, with demand d_i and loading time t_i, $\forall i \in N$.
- The number of vehicles available at the depot, V.
- The distance matrix, D, with the pairwise distances D_{ij} from node i to node j.
- The time matrix, T, with pairwise times T_{ij} to go from node i to node j.
- The maximum capacity of each vehicle, Q.
- The maximum duration allowed for a route L.

Note that D and T are input data, and not necessarily symmetric. A standard way to model vehicle routing problems is with the so-called set partitioning formulation, which is based on the definition of R, the set of routes that satisfy the problem requirements. In our case, it means that each route in R must start and finish at the depot, not to exceed the capacity (Q) and duration (L) limits, and visit at most once a customer.

To model the problem in mathematical terms, we define the binary decision variable x_i that is equal to 1 if route $i \in R$ is selected (and therefore a vehicle visits the customers in that route). The set partitioning formulation of the standard, single –objective problem minimizing the total distance, is then:

$$\text{Min} \sum_{i \in R} x_i D_i$$

s.t.:

$$\sum_{i \in R} x_i a_{ij} = 1 \;\; j \in N$$

$$\sum_{i \in R} x_i \le V$$

$$x_i \in \{0, 1\} \;\; i \in R$$

The objective function minimizes the total distance of the selected routes, where D_i is the distance of route i, computed from the pairwise distances D_{ij}. In particular, if a route i sequentially visits nodes $i_1, i_2, \dots,\; i_s$, then $D_i = D_{i_1 i_2} + D_{i_2 i_3} + \dots + D_{i_s i_1}$. Similarly, we can compute the travel time T_i of route i as $T_i = T_{i_1 i_2} + T_{i_2 i_3} + \dots + T_{i_s i_1}$, its associated loading time $lt_i = t_{i_1} + t_{i_2} + \dots + t_{i_s}$, and the route demand (truck capacity required to serve the route) as $d_i = d_{i_1} + d_{i_2} + \dots + d_{i_s}$.

The two main implicit constraints of our model are related to the demand, which is restricted to the truck capacity to pick-up the waste (Q), and the time limit of the routes given by the shifts of the drivers (L). The definition of the set R assumes that its routes satisfy these constraints, so in mathematical terms:

$$R = \{ feasible\ routes\ i : \;\; T_i + \; lt_i \le L, \;\; d_i \le Q \; \}$$

The first group of explicit constraints in the formulation impose that each node is in exactly one selected route, defined from the a_i vectors, where $a_{ij} = 1$ if route i visits node j. The second constraint limits the selection of routes to the number of vehicles V.

Thus, the objective functions f_1 and f_2 can be formulated from these elements as follows:

$$f_1 = \sum_{i \in R} \; x_i D_i$$

$$f_2 = \max_{i \in R} \; x_i D_i$$

A straightforward formulation of f_3 is given by the expression $\max\limits_{i \in R: x_i = 1} T_i - \min\limits_{i \in R: x_i = 1} T_i$, which can be modeled introducing an auxiliary variable s and a big M constant as:

$$f_3 = \max_{i \in R} \; x_i T_i max\ s$$

$$s \le x_i T_i + M(1 - \; x_i) \;\; i \in R$$

where M is a larger value than the T_i, for example the maximum of these values.

In line with previous studies (Pacheco & Martí, 2006), the number of routes (f_4) is considered in a special way when modeling and solving the problem. Due to the availability on the number of vehicles and drivers shifts, f_4 usually may just take a few values. A standard way, known as the epsilon-constraint method, to deal with objectives that are limited to take a small number of values, as it is the case of f_4, is to include the objective as a problem constraint, and then iteratively solve the problem for every value of the right-hand-side of this constraint.

Note that our model does not explicitly define time windows within the service, but according to the real application, it has a limitation of the length, in terms of time, of the driven routes (L). The variation of the number of routes (f_4) will allow a future decision marker to incorporate multi-trips to the collection plan.

The contribution of our proposal to the extensive literature summarized in Section 2 mainly comes from the consideration of route balancing in terms of both distance and time. Analyzing the balance of routes is an interesting objective lately considered in order to state similar working conditions among all the worker that perform this kind of service. In the literature, this objective has been tackle using different definitions, see for example Jozefowiez, Semet and Talbi (2009) or Oyola and Løkketangen (2014). Its relevance is mainly due to the real application, where there are distance and time constraints related to the worker's shifts. In this context, it is important to design balance routing systems, so that all the workers perform similar loading works.

The simultaneous consideration of four objectives permits to model a realistic version of the problem, but it is clear that it also brings complexity to the model. In order to simplify the present document, and based on the descriptions of similar transportation problems (Jacobsen & Madsen, 1980), we do not provide a formal formulation of the model and we refer the reader to the many previous papers described above where he or she may find similar formulations. To perform an efficient search in our solving method, we include Wierzbicki's function to guide our metaheuristic method in such a complex context. As described in the following section, the combination of both elements together lead to a novel approach, which provides high quality solutions.

4. Methodology

In this paper, we follow a standard approach to deal with multiple objectives, the so-called achievement scalarizing function (ASF) (Miettinen, 1999; Wierzbicki, 1980), which has the ability to produce any Pareto optimal or weakly Pareto optimal solution. It is based on the idea that we want to obtain efficient solutions in the direction of the reference point. If this point is actually the utopia point, also called ideal point, we try to obtain solutions as close as possible to it, by minimizing the value at ASF considering this reference point.

Let f_i^{min} and f_i^{max} be the minimum and maximum values of $f_i(x)$ respectively over all feasible solutions x. It must be noted that, since we are proposing a heuristic method and we cannot effectively enumerate all the feasible solutions of our problem, we do not know these exact values and we are just computing them over the set of solutions known (i.e., generated so far in the search process). These values provide a standard way to define a reference point (although any other values could be used for that purpose as well). Since we are minimizing the k objectives, we use the utopia point given by $(f_1^{min}, f_2^{min}, \dots, f_m^{min})$ as the reference point in the search, though another one could have been used as well.

We consider one of the most applied ASF, the Wierzbicki's achievement function (Wierzbicki, 1980). In general, any ASF minimizes the distance from a reference point, usually given by the decision maker, to the feasible region if the reference point is unattainable. In particular, Wierzkbicki's achievement function modifies the Tchebyschev distance to ensure the generation of efficient solutions. It has been applied in many multiobjective problems, for example in Stam, Kuula and Cesar (1992) to tackle an acid rain problem in Europe by an interactive decision making process. Considering that the differences in magnitudes might cause a bias in the evaluation, this function normalizes the input values. In mathematical terms, given a solution x, its Wierzbicki's value $W(x)$ is computed as:

$$W(x) = \max_{i=1,\dots,m} \left\{ \lambda_i \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} \right\} + \rho \cdot \sum f_i(x)$$

Given a set of λ_i-values, reflecting the importance or user preference of each objective, the optimization of Wierzbicki's function provides the closest efficient solution to the reference point according to those values. It is clear that different values of these parameters (λ_i) lead to different solutions, so we propose to perform multiple searches varying these parameters to obtain a good approximation of the entire Pareto front.

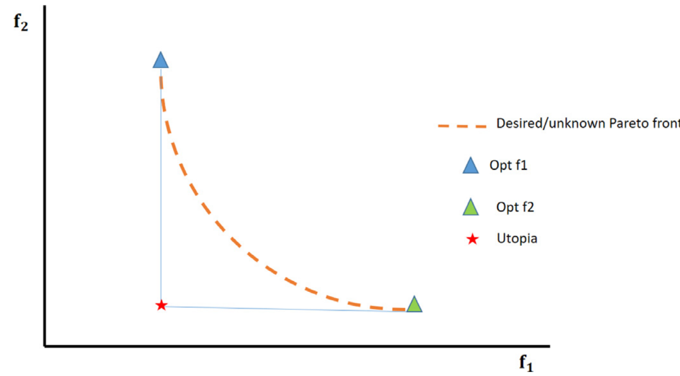

Fig. 2. Illustration of the ideal or utopia point.

Fig. 2 shows an illustration of the utopia point (f_1^{min}, f_2^{min}) in a bi-objective case. Note that this point does not represent a solution to the problem, and can be considered as an aspiration point.

In a general multiobjective optimization problem (1), the utopia point is defined as $z^* = (z_1^*, z_2^* \dots z_m^*)$, where $z_i^* = \min f_i(x)\ \forall i = 1, 2 \dots m$. In simple words, the utopia point is the one with components the best values for each objective function in the Pareto optimal set. In the next sections, we describe a heuristic method to optimize the Wierzbicki's function $W(x)$ described above. It first performs single-objective searches to identify good values of each objective, in order to compute the utopia point. Then, it minimizes $W(x)$ as a strategy to obtain different solutions of the multiobjective problem, thus generating an approximation to the Pareto front. Hence, based on these assumptions, the larger the number of iterations varying this set of λ_i-values, the more accurate the approximation is expected to be. However, the computational cost will increase, so the process is divided into two stages, where the second seeks to improve the density of the first approximation of the Pareto front, as detailed in the following section. This function is integrated in our solving method described in the next section. It will be included in an expert system, to help decision makers to obtain a good solution according to the problem characteristics.

5. Solving method

Following basic Tabu Search (TS) principles (Glover & Laguna, 1997), memory structures can be implemented within a constructive process to include certain elements in the solution previously identified as attractive. This leads to a different perspective in constructive methods than the traditional building from scratch, applied in classic multi-start methods such as GRASP (Feo & Resende, 1995). Although it is commonplace in the metaheuristic literature to restrict the word "neighborhood" to refer solely to transitions between solutions as embodied in improvement methods, within this new perspective based on memory strategies, constructive neighborhoods may play an important role.

Constructive and destructive neighborhoods have been applied within an effective method known as iterated greedy (IG, Ruiz & Stützle, 2007)), which generates a sequence of solutions by iterating over a greedy constructive heuristic which uses two main phases. It starts from a complete initial solution, and then iterates through a main loop, where the method first generates a partial candidate solution by removing a fixed number of elements from the complete solution (Destruction phase), and next reconstructs a complete solution (Construction phase) by adding elements to the partial candidate solution. The method is coupled with a local search phase, where an improvement procedure is performed to obtain a local optimum (in case of mono-objective optimization).

Algorithm 1 General framework for Iterated Greedy.

IteratedGreedy(S, it, β)
1. **for** $i \in 1 \ldots it$ **do**
2. $S' \leftarrow Destruct(S, \beta)$
3. $S'' \leftarrow Reconstruct(S')$
4. $S \leftarrow AcceptanceCriterion(S, S'')$
5. **end for**

Algorithm 1 presents the pseudocode for the general Iterated Greedy framework. The method requires three input parameters: S, the initial solution; it, the maximum number of iterations performed; and β, the percentage of solution to be removed (destroyed) in the first stage of the algorithm. Although in general terms the initial solution can be generated at random, in line with previous works, we consider a constructive procedure to provide IG a good initial point to start the search to obtain better results and to reduce the time required to reach the best solution. In particular, we propose a greedy randomized constructive method to generate the initial solution for the Iterated Greedy algorithm, which is described in Section 4.1.

The algorithm performs it iterations (steps 1–5 in Algorithm 1), where each iteration consist of two different stages: destruction and reconstruction. The former is devoted to remove a percentage (β) of elements of the incumbent solution, while the latter is designed to obtain a new feasible solution by reinserting the previously removed vertices in different positions. Each iteration is performed with a set of random directions (selections of the λ_i values), defined in terms of $\{f_i^{min}, f_i^{max}\}_{i=1,\ldots,4}$. Here, each solution obtained during the construction is checked for its inclusion into the *non-dominated solutions* set, or otherwise, into the *promising solutions* set if they are within a percentage Δ of the efficient solutions, called Δ-efficient region. Fig. 3 represents an illustrative example with 2 objectives of this region where the promising solutions are located.

In this paper, we propose to apply a local improvement step based on Variable Neighborhood Search (VNS) to improve the solutions obtained with the Iterated Greedy algorithm in each destruction-reconstruction iteration. The neighborhoods used in this improvement phase are presented in Section 5.3, while the VNS approach is described in Section 5.4.

It is worth mentioning that any non-dominated solution found during the search, in any of the steps of our method, is checked for its possible inclusion in the Pareto front. Also note that the improvement method (VNS) seeks to find new non-dominated solutions by optimizing the distances from the promising solutions to a reference point with components defined by the best values obtained between two different solutions at each coordinate, i.e. $Ref = (f_1^{min}, f_2^{min}, f_3^{min}, f_4^{min})$.

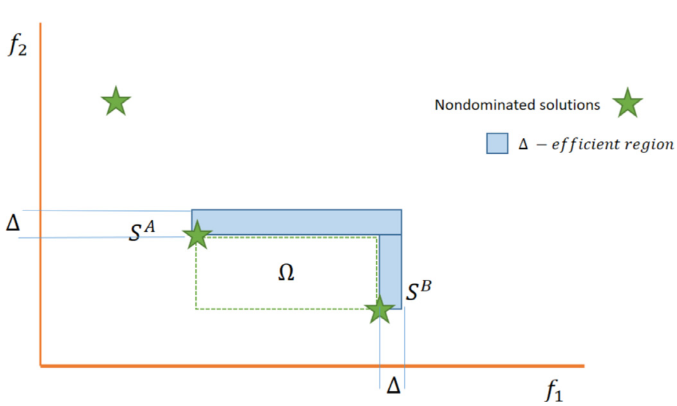

Fig. 3. Δ-efficient region example.

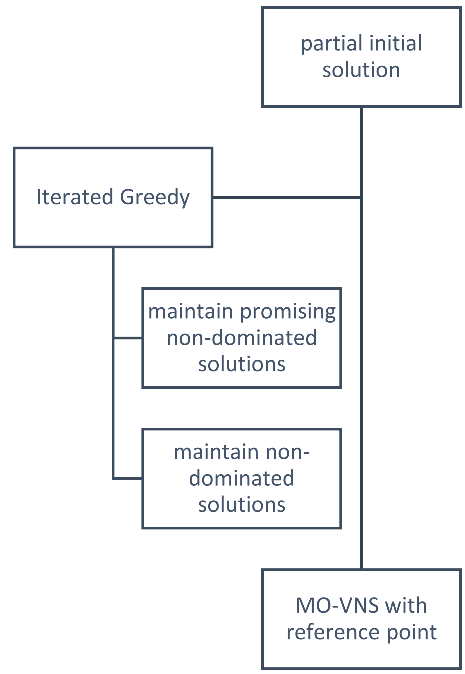

Fig. 4. Algorithm scheme.

The proposed algorithm is designed to be included in an expert system as a black box. In particular, the system will provide the algorithm with a set of inputs, indicating the location of the bins and depot (so it can determine the distance and time matrix), the amount of waste expected at each point, and the time required emptying the waste at each stop. Then, the algorithm will return the set of non-dominated solutions. After that, the expert will be able to select the non-dominated route, from those provided by the system, that better adapts to the problem under consideration. Fig. 4 represents a scheme of the algorithm functionality:

5.1. Initial solution

As mentioned in the introduction, objective f_4 (number of vehicles) takes a limited number of values, and therefore we use an analogous approach to the eps-constraint Haimes, Lasdon and Wismer (1971) where f_4 will play the role of a bounded constraint so that our search method optimizes the first three objectives for a particular value of f_4. We therefore compute the Wierzbicki's function $W(x)$ with f_1, f_2 and f_3. In particular, we compute the lower bound for the number of vehicles f_4^{min} as the sum of the demands divided by the capacity of the vehicle. From this initial value, we consider a fixed number of consecutive integer values until we reach f_4^{max}, which is set beforehand in our experiments.

For a given value v of f_4, our constructive method starts by randomly selecting a set of "seed" customers $i_1, i_2, i_3, \ldots, i_v$, which are assigned to vehicles 1, 2, …v, respectively. Then, for each of them we create a close tour from the depot to customer i_j, creating in this way the initial partial solution x_0. We compute the Wierzbicki's function in this initial partial solution, $W(x_0)$, as a baseline to evaluate the candidate nodes to complete it. At this stage, the set of candidate nodes C is formed with all the non-assigned nodes.

For each candidate node, $s \in C$, we compute the Wierzbicki's function value corresponding to the partial solution if it is inserted in each tour, and select the best tour t_s^1 and the second best tour t_s^2 for insertion according to the W-value (where the lower the value, the better the tour). The *extra-mileage value* criterion (Mole & Jameson, 1976) selects the candidate node s^* with minimum

W-value and inserts it in tour t^1_{s*}. The *regret value* (Fisher & Jaikumar, 1981) computes the difference between extra-mileage values corresponding to the insertion of node s in tours t^2_s and in t^1_s, and selects the node s^* with a maximum difference and inserts it in t^1_{s*}.

The construction of the solution is guided by both criteria, the extra-mileage and the regret function. In particular, a random value μ ∈ [0, 1], is generated to determine the proportion of nodes to be inserted according to the maximum value of the regret function. Then, to complete the solution, additional nodes are inserted according to the minimum extra – mileage value. To reduce computational cost, after inserting a node s^* in tour t^1_{s*}, it is removed from the candidate list C and, only, the extra - mileage value is updated for each $s \in C$ in tour t^1_{s*}. In line with this, the W-value is determined by estimating the value of each objective function if node s is inserted in tour t_i, and the best position (p) for each tour is also saved for each tour of the current solution. The method finishes when all the customers (nodes) have been inserted.

5.2. Iterated greedy method

Destruction and reconstruction stages must be designed in order to balance diversification and intensification during the search. Considering traditional implementations of Iterated Greedy, diversification takes place in the destruction phase, randomly removing some elements from the solution, while intensification relies on the reconstruction phase, inserting the removed elements again in the solution following a certain greedy criterion.

We propose two destructive methods and two constructive methods to study the influence of diversification and intensification in the quality of the obtained solutions. Specifically, we propose both random and greedy approaches respectively for each stage.

The destruction consist of selecting $\beta \times n$ nodes from the solution, removing them from their corresponding routes. In the *random variant*, we select the nodes to be removed in an arbitrary manner, thus increasing the diversification of the search. In the *greedy variant*, the destruction removes the nodes with the largest contribution to the Wierzbicki objective function. In particular, all the nodes are sorted in descending order with respect to its contribution to the Wierzbicki function value. Then, the first $\beta \times n$ nodes are removed from the solution.

The reconstruction phase is responsible of re-inserting the nodes that have been previously removed from the solution in the destruction phase. Again, we propose two variants of this phase: random and greedy. The *random variant* inserts each removed node in a random position on a randomly selected route. It is worth mentioning that the method selects a random route among those with enough capacity to host the current node, always producing a feasible solution. The *greedy variant* sorts the removed nodes in ascending order with respect to its contribution to the Wierzbicki function (i.e., those which produces smaller values of the objective function are selected first). Specifically, the contribution of each node is evaluated as the Wierzbicki function value when inserting the node in the route and position that produces the minimum Wierzbicki function value.

Combining the random and greedy variants with the destruction and reconstruction methods, we consider four different variants for the Iterated Greedy procedure (shown in Table 2). A discussion on the performance of each variant is presented in the computational experiments reported in Section 5.

Table 2
Iterated Greedy variants.

Algorithm	Destruction	Reconstruction
IG_{rr}	random	Random
IG_{rg}	random	Greedy
IG_{gr}	greedy	Random
IG_{gg}	greedy	Greedy

5.3. Local search

The solution obtained by applying the destruction and reconstruction phase can be improved by using a local search method in order to obtain a local optimum with respect to a certain neighborhood. Note that in multiobjective optimization, the exploration of the neighborhood during the local search stage can identify new non-dominated solutions, which are included in the set of non-dominated solutions (sometimes called *the archive* in the literature). Specifically, we consider three classic neighborhoods for the local search phase: insert, exchange, and 2-opt.

The *insert neighborhood* consists in all the solutions that can be reached by performing a single insertion move in any of the routes. Specifically, the insertion move *insert*(S, r_s,r_d,v, i) removes node v from its current route r_s and inserts it on route r_d in position i.

The *exchange neighborhood* contains all the solutions that can be obtained by swapping two nodes in the same or in different routes. The move *exchange*(S, v_1,r_1,v_2,r_2) consist of removing node v_1 from route r_1, inserting it at the position assigned to node v_2 in the route r_2 (and symmetrically for node v_2).

Finally, the *2-opt neighborhood* contains all the solutions that can be obtained by performing a single 2-opt move to the incumbent solution. The 2-opt move is a common move in vehicle routing problems (Potvin & Rousseau, 1995), and consists in reversing a certain sub-chain in a route. The 2-opt neighborhood presented in this work contains all the solutions that can be obtained by reversing a certain part of a route, considering that the number of elements reversed can vary from 2 to the total number of nodes in route. We do not consider a reversal of size 1 since it would not make any change in the solution. In the next subsection we describe how we implement these three neighborhoods in a variable neighborhood search exploration.

5.4. Variable neighborhood search method

The iterated greedy process generates a set of nondominated solutions, while it maintains a pool of promising solutions (PS for short). Now, a new local search procedure is run to find more efficient solutions by minimizing the distances of elements in PS to a reference point. Although a simple local improvement can be considered here, we apply a Variable Neighborhood Search (VNS) algorithm to obtain even better outcomes.

VNS is a metaheuristic framework originally proposed by Mladenovic and Hansen (1997) with the aim of solving hard optimization problems. The VNS framework relies on the idea of performing systematic changes of neighborhood during the search for exploring different neighborhood structures, thus increasing the portion of the search space explored. Since the original VNS algorithm was proposed, several variants have been designed. For instance, we can highlight Basic VNS, Variable Neighborhood Descent (VND), Reduced VNS, General VNS or Variable Formulation Search, among others (see: Hansen, Mladenović, Todosijević & Hanafi, 2017; Pardo, Mladenovic, Pantrigo & Duarte, 2013) for a complete review of these variants). We apply GVNS in our method.

GVNS combines deterministic and random changes of neighborhoods, balancing intensification and diversification during the search. Algorithm 2 shows the pseudocode of GVNS where we can see that the algorithm starts from the first considered neighborhood (step 3). Then, GVNS iterates until reaching the maximum predefined neighborhood k_{max}, which is the single parameter of the algorithm. For each iteration, the *shake* method generates a

Algorithm 2 GVNS pseudocode.

1. *GVNS*(S, $k_{\max}$)
2. *repeat*
3. $k \leftarrow 1$
4. *while* ($k \leq k_{\max}$)
5. $S' \leftarrow shake(S, k)$
6. $S'' \leftarrow VND(S')$
7. $k \leftarrow NeighborhoodChange(S, S'', k)$
8. *endwhile*
9. *until Stopping Criterion*

random solution S' in the neighborhood N_k of the incumbent solution S (step 5). Then, solution S' is improved with the Variable Neighborhood Descent (VND) algorithm until reaching a local optimum S'' with respect to the considered neighborhoods (step 6). Finally, GVNS checks whether the local optimum S'' is better than the incumbent solution S (step 7), updating the best solution ($S \leftarrow S''$) and restarting the search from the first neighborhood ($k \leftarrow 1$), or not, continuing the search in the next neighborhood ($k \leftarrow k+1$). The algorithm stops when reaching the maximum neighborhood to be explored k_{max} without finding any further improvement. These steps are repeated until a certain stopping criterion is found, usually a predefined number of iterations or a maximum computing time. Since in this work VNS is used as an improvement method, we want to keep its running time moderate, so we only consider a single iteration of GVNS.

As shown in Section 4.3, we consider three different neighborhoods to explore the solution space: insert (N_1), exchange (N_2), and 2-opt (N_3). The VND procedure systematically explores these three neighborhoods until reaching a local optimum with respect to all of them. Algorithm 3 presents the pseudocode of VND. The method

Algorithm 3 VND pseudocode.

1. *VND*(S, $N_1 \ldots N_{max}$)
2. $k \leftarrow 1$
3. *while* ($k \leq k_{\max}$)
4. $S^{\star} \leftarrow argmin_{S' \in N_k} f(S')$
5. $k \leftarrow NeighborhoodChange(S, S', k)$
6. *endwhile*

starts by exploring the first neighborhood (step 2), iterating until reaching the maximum considered neighborhood (steps 3–6). In each iteration, VND finds the best solution $S^{\star}$ in the current neighborhood (step 4). Note that $f(S)$ is the objective function value of solution S. Then, if $S^{\star}$ outperforms the incumbent solution S, it is updated ($S \leftarrow S^{\star}$), restarting the search from the first neighborhood ($k \leftarrow 1$). Otherwise, the method explores the next neighborhood ($k \leftarrow k+1$). VND ends when no improvement is found in any of the neighborhoods.

It is well documented that in VND the order in which the neighborhoods are explored can make a difference. After preliminary experimentation, we select to explore the neighborhoods in the order: $N_2 \rightarrow N_3 \rightarrow N_1$.

After the application of the Iterated Greedy with the GVNS improvement method described above, we apply a final post-processing. In particular, VNS is also used to further improve the approximation of the Pareto front obtained so far. In particular, we apply the GVNS described in Algorithm 2 to each solution S in this front, to minimize the L_2 – distance between S and the reference point *Ref* described above. As it is customary in multiobjective optimization, all non-dominated solutions found in this exploration are submitted for inclusion to the Pareto front approximation, which is returned as the output of the method.

To sum it up, the complete algorithm starts from an initial solution generated as described in Section 5.1. After that, the IG method iteratively destructs and reconstructs the solution until no further improvement is possible. Once IG is finished, the VNS is applied to the set of promising non-dominated solutions, with the aim of populating those regions of the Pareto Front that IG has not been able to cover. This approach is represented in Fig. 4.

Table 3
Parameters on instance generator.

D_{ij}	sp	Tr
$D_{ij} < 200$	10	1
$200 \leq D_{ij} \leq 600$	20	0.8
$D_{ij} > 600$	30	0.6

6. Computational experiments

In this section we first analyze the performance of our algorithm and then compare it with the well-known methods NSGA-II (Deb, Pratap, Agarwal & Meyarivan, 2002) and SPEA2 (Zitzler, 2003) on a set of public domain instances. In particular, we consider the VRP-lib,[1] which consists of a set of 100 instances with sizes ranging between 100 and 1000 customers recently introduced by Uchoa et al. (2017). Since these instances do not include time constraints, we have created a time matrix for each instance simulating a real situation. To do so, we take into account the relationship between time, speed and distance, including a bias to consider traffic variations. Hence, to generate a realistic time between customers i and j, denoted as T_{ij}, we use the following expression:

$$T_{ij} = \frac{D_{ij}}{sp} \cdot \frac{1000}{3600} \cdot Tr$$

Note that the distance matrix (D_{ij}) is determined by the Euclidean distance between each pair of customers i and j, with $i, j \in \{1, \ldots n\}$. Additionally, to generate T_{ij} we use the values for speed (sp) in km/h and traffic (Tr) in Table 3.

Table 3 details the parameter values applied in our study to generate a more realistic time matrix. In particular, studying the road specifications and driving rules in Spain, as well as their conditions in our case study, we consider that for a distance under 200 m, the speed should not be larger than 10 m/s and a light traffic (Tr) would be found on the road. However, for customers located among 200 and 600 m' distance, we assume a normal speed of 20 m/s for a truck and we introduce a small perturbation in the traffic. Finally, for those containers located further than 600 m, the speed might be larger, but the traffic would be relatively heavy.

The experiments are divided into two parts: (1) Preliminary experiments for tuning the parameters of the algorithm and (2) Final experiments to test the performance of the algorithm. The former considers a subset of 25 representative instances out of the 100 to avoid overfitting. All the experiments have been performed using an Intel Xeon CPU E3-1240 v6 3.7 GHz CPU and 8 GB RAM.

In the final experiments we compare our method with NSGA-II and SPEA2. We adapt these well-known solvers to the waste routing problem This adaptation basically consists in two parts: the codification of the solutions and the creation of the initial population of solutions. Regarding the codification of solutions, we have considered a classical approach with a chromosome divided into two parts. The first part of the chromosome stores the sequence of customers that are visited, and the second part of the chromosome stores the number of customers of the sequence that correspond to each one of the vehicles. Under this approach, the crossover operator is applied to the first part of two parent individuals,

[1] http://www.vrp-rep.org/datasets/item/2016-0019.html

maintaining the permutation structure, while mutation is applied to the second part of the chromosome.

To create a competitive implementation of these two methods, we generated the individuals in the initial population by randomly selecting a constructive method for each one of the solutions. The constructive method tries to improve only one of the objectives, following the heuristic method described in Section 4.1. Applying a uniform distribution in the selection, the initial population contains a similar number of individuals created according to each one of the objectives in our study. The methods were implemented in Java using jMetal (Durillo & Nebro, 2011).

6.1. Preliminary experimentation

The proposed algorithm requires of the analysis of 3 parameters: (1) the maximum neighborhood to be explored in the VNS, *kMax*; (2) the percentage of solution that will be removed in the Iterated Greedy, β; and (3) the order of random and greedy stages inside the Iterated Greedy. In particular, we consider the following values for each experiment:

1 $kMax = \{0.1, 0.25, 0.5, 0.6\}$.
2 $\beta = \{0.05, 0.1, 0.15\}$.
3 $IG = \{IG_{rr}, IG_{rg}, IG_{gr}, IG_{gg}\}$.

A full factorial experiment would imply the evaluation of all the possible combinations, resulting into a 3 • 3 • 4 = 36 different experiments, which would be impractical in terms of computing times. However, as suggested in Sánchez-Oro, Laguna, Martí and Duarte (2016), a sequential design of the preliminary experimentation produces similar results. We first set the *kMax* value, then the β parameter and, finally, the best Iterated Greedy variant.

Comparing two or more algorithms in MOP is a difficult task. There are a set of quality indicators to analyze and compare the performance of MOP by means of evaluating different aspects of the Pareto front generated by each algorithm (Zitzler, Thiele, Laumanns, Fonseca & Da Fonseca, 2003). We report the results in the following experiments with three of the most relevant metrics used to evaluate the performance of multiobjective algorithms: Coverage (Zitzler, 1999), hypervolume (Zitzler & Thiele, 1998) and epsilon metric (Zitzler et al., 2003). Prior to evaluate these metrics, we need to create the *reference set*, formed by the union of all the approximations of the Pareto set to be evaluated. In other words, the *reference set* will contain those solutions that are non-dominated among all the solutions generated by every algorithm. This *reference set* is our best approximation to the optimal Pareto front, considering that it is not known.

Given the Pareto front generated by an algorithm, *P*, and the aforementioned *reference* set, *RS*, the *coverage metric (CV)* provides the percentage of solutions of *P* that are dominated by one or more solutions belonging to the *RS*. Therefore, the lower the coverage value, the better the quality of the analyzed Pareto front. This value can be obtained by the following formula:

$$C(P, RS) = \frac{|\{b \in P : \exists a \in RS \text{ such that } a \preccurlyeq b\}|}{|P|}$$

where $a \preccurlyeq b$ denotes that “a dominates b”. Also, the *Hypervolume* (HV) calculates the volume (size of the portion) in the objective space that is covered by the Pareto front under evaluation. Then, the larger the value, the better the Pareto front. The *Epsilon* (EPS) measures the smallest distance required to move every point of the Pareto front until it dominates the reference Pareto front. Regarding this metric, a small value indicates a better Pareto front.

Table 4 shows the solution of our first preliminary experiment. Specifically, it shows the average values obtained for the three aforementioned metrics obtained for each value of *kMax*= 0.1*n*, 0.25*n*, 0.5*n*, and 0.6*n*. Analyzing the results, we can see the superiority of $kMax = 0.5n$, which is able to outperform in every metric the other variants, with the exception of $kMax = 0.6n$ that provides similar results in terms of quality but in longer running times (1431.75 s on average with $kMax = 0.5n$ versus 1864.6 s with $kMax = 0.6n$). Additionally, we can see that low values (first rows in Table 4) present similar results, which lead us to conclude that a considerably large perturbation inside the shake method is required to improve the quality of the solutions generated. We therefore set $kMax = 0.5n$ for the rest of the experiments.

Table 4
Average value of MOP quality indicators.

kMax	CV	HV	EPS
0.10*n*	0.5498	0.7244	0.1182
0.25*n*	0.5479	0.7297	0.1159
0.50*n*	**0.5290**	**0.7407**	**0.0953**
0.60*n*	0.5100	0.7400	0.0953

Table 5
Average value of MOP quality indicators for beta.

β	CV	HV	EPS
0.05	0.5948	0.7531	0.0974
0.10	0.5804	**0.7577**	**0.0973**
0.15	**0.5351**	0.7526	0.1074

In the second experiment we study the parameter β. This parameter indicates how many nodes are removed from the solution in the destruction phase within Iterated Greedy. In this context, three different values of β are considered in Table 5. It is worth mentioning that a large destruction of the incumbent solution is not recommended in the Iterated Greedy literature, since it would be equivalent to constructing a complete solution from scratch, without maintaining the features of the original solution. Results in this table show that $\beta = 0.1$ presents better values in two out of the three 3 metrics studied, so we select this value in the next experiments.

In our third preliminary experiment we undertake to explore the destruction and reconstruction criterion between random and greedy in the Iterated Greedy algorithm. As described below, four different variants are considered: Random-Random, IG_{rr}; Random-Greedy, IG_{rg}; Greedy-Random, IG_{gr}; and Greedy-Greedy, IG_{gg}. These variants are now compared, considering the best values in the previous experiments: $kMax = 0.5$ and $\beta = 0.1$. Table 6 shows the associated results.

Results in Table 6 show that IG_{gg} and IG_{gr} are the best variants. As the reader may observe, IG_{gr} only obtains a better value in the Hypervolume indicator, while IG_{gg} achieve better results in two over the three indicators considered. However, the Greedy-Greedy variant lacks diversification, resulting in very similar solutions. Considering that VNS will be applied to further improve the obtained Pareto front, we have selected the Greedy-Random variant to increase diversity, even if it is better in just one of the metrics, since it will allow the final algorithm to explore a wider portion of the search space.

Table 6
Average value of MOP quality indicators to contrast IG.

IG	CV	HV	EPS
IG_{gg}	**0.5751**	0.7639	**0.1074**
IG_{gr}	0.6260	**0.7645**	0.1081
IG_{rg}	0.6907	0.7590	0.1154
IG_{rr}	0.6532	0.7598	0.1124

Table 7
Comparison of MOP, SPEA2 and NSGA-II.

small (100–199)	CV	HV	EPS
*IG*_GG_VNS	**0.4401**	**0.7462**	**0.1037**
*IG*_GR_VNS	0.5414	0.7459	0.1226
NSGA-II	0.8113	0.6480	0.2033
SPEA2	0.7847	0.6467	0.2030
medium (200–399)			
*IG*_GG_VNS	**0.4450**	**0.7851**	0.1012
*IG*_GR_VNS	0.4979	0.7789	**0.0963**
NSGA-II	0.7943	0.6799	0.1929
SPEA2	0.7933	0.6787	0.1941
large (400–1000)			
*IG*_GG_VNS	0.5093	0.7734	0.1269
*IG*_GR_VNS	**0.5013**	**0.7790**	**0.0958**
NSGA-II	0.7906	0.6710	0.2306
SPEA2	0.7788	0.6735	0.2266

6.2. Competitive testing

The specific problem considered in this paper has not been addressed by heuristic or exact methods in the literature, so to analyze the relative quality of our proposal, we have considered the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) and the Strength Pareto Evolutionary Algorithm (SPEA2), two state-of-the art multiobjective methods for generating high-quality Pareto fronts. Table 7 shows the comparison of the two best configurations of our IG_VNS method ($kMax = 0.5$, $\beta = 0.1$, $IG_{gg}$and IG_{gr}) with these two methods, NSGA-II and SPEA2. The parameter values for these algorithms have been selected after some preliminary experiments starting from their standard configuration. For NSGA-II we tested three typical values for crossover and mutation probabilities: 0.5, 0.7, 0.9 and 0.1, 0.2 and 0.3, respectively. These values were run on a reduced set of representative instances fixing the population size to 250 individuals for three different number of generations: 5000, 10,000 and 15,000. Once the parameter values were selected for NSGA-II, we used the very same values for SPEA2 in order to use the same number of evaluations. We use the same subset of 25 representative instances that in the calibration of IG_VNS. Their values are the following:

- Population size: 250
- Generations: 10,000
- Single-Point crossover with probability = 0.75
- Uniform mutation with probability = 0.2
- Binary tournament selection

As in the previous experiments, we report the Coverage (CV), hypervolume (HV), and epsilon metric (EPS). Table 7 reports the results over the entire set of 100 instances (Uchoa et al., 2017) divided according to their size.

Table 7 clearly shows that the proposed algorithms are able to outperform both methods, NSGA-II and SPEA2, according to the three metrics considered. In particular, IG_GG_VNS presents the best results in small instances, while IG_GR_VNS is the clear winner in the large ones. Considering the coverage value (CV), our two methods obtained values significantly lower than NSGA-II and SPEA2 in the three sets of instances. For example, in the medium size instances, the ranking of the methods according to CV is IG_GG_VNS (0.4450), IG_GR_VNS (0.4979), SPEA2 (0.7933), and NSGA-II (0.7943). This indicates that less than a 50% of the Pareto fronts generated by our methods are covered by solutions in the reference set, while almost 80% of the solutions in the Pareto fronts of the previous methods are covered by reference solutions. Analyzing the hypervolume (HV), we obtain a similar ranking of the methods, and we can clearly see that the values for the proposed algorithms are larger than those for NSGA-II and SPEA2, indicating that their approximations to the Pareto front cover a larger volume than those covered by the other methods. Finally, the small epsilon value (EPS) for the proposed algorithms indicates that they are very close to the reference set. This result means that, even in those solutions that are covered by the reference solutions, the ones generated by the proposed algorithms are close to the reference ones.

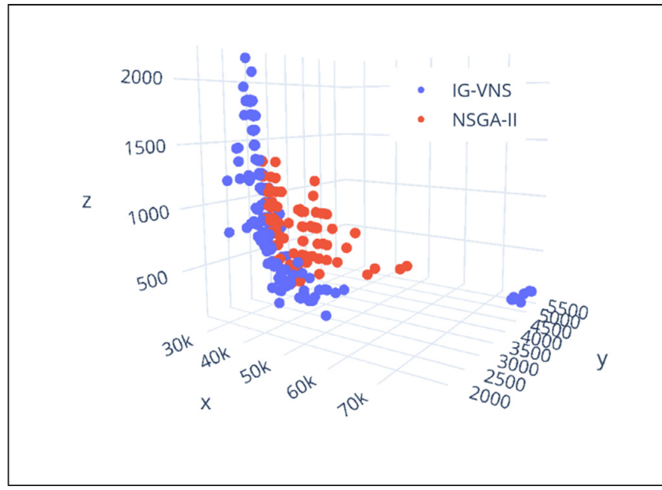

Fig. 5. Approximations of the Pareto front.

To draw significant conclusions, we perform a statistical test with the data reported in Table 7. In particular, we perform the non-parametric Wilcoxon test, which results in a *p*-value lower than 0.00001, which confirms that there are significant differences between the methods tested.

To complement the quantitative information above, we represent now the approximation of the Pareto front obtained with the two methods on a representative instance. In particular, Fig. 5 represents, for each non-dominated solution obtained with each method, a point with coordinates corresponding to f_1, f_2, and f_3, while keeping the number of routes, f_4 fixed. In particular, the X and Y axis, which define the bottom surface, is determined by the values corresponding to f_1 and f_2 in meters, and the Z axis represents the values of f_3 in seconds.

It is clear from the graphic shown in Fig. 5 that the IG_VNS method obtains a better set of points than the NSGA-II, since it contains more points and most of them dominate the points obtained by NSGA-II. This visual conclusion is reinforced by the results obtained in the tables described above.

To study the influence of the number of routes, given by f_4, in the approximation of the Pareto front obtained with our method, we depict in Fig. 6 different sets of solutions corresponding to different values of this objective on a representative instance. Specifically, solutions with the same number of routes f_4 are plotted using the same color, where f_4 ranges from 43 to 47. As expected, we can see that the values of these three objectives highly depend on the number of routes that we are considering, thus obtaining very different solutions.

7. Case study

Currently, the waste collection service in the region of Antequera is based on a frequent routing systems. The increasing amount of waste generated in the last years have encouraged the administrations to evaluate the cost incurred to a daily service, attending to different constraints and goals. In particular, the routing system within this region has not previously been optimized, so providing the managers with a set of alternatives according to their requirements is the main motivation of this work. In addition

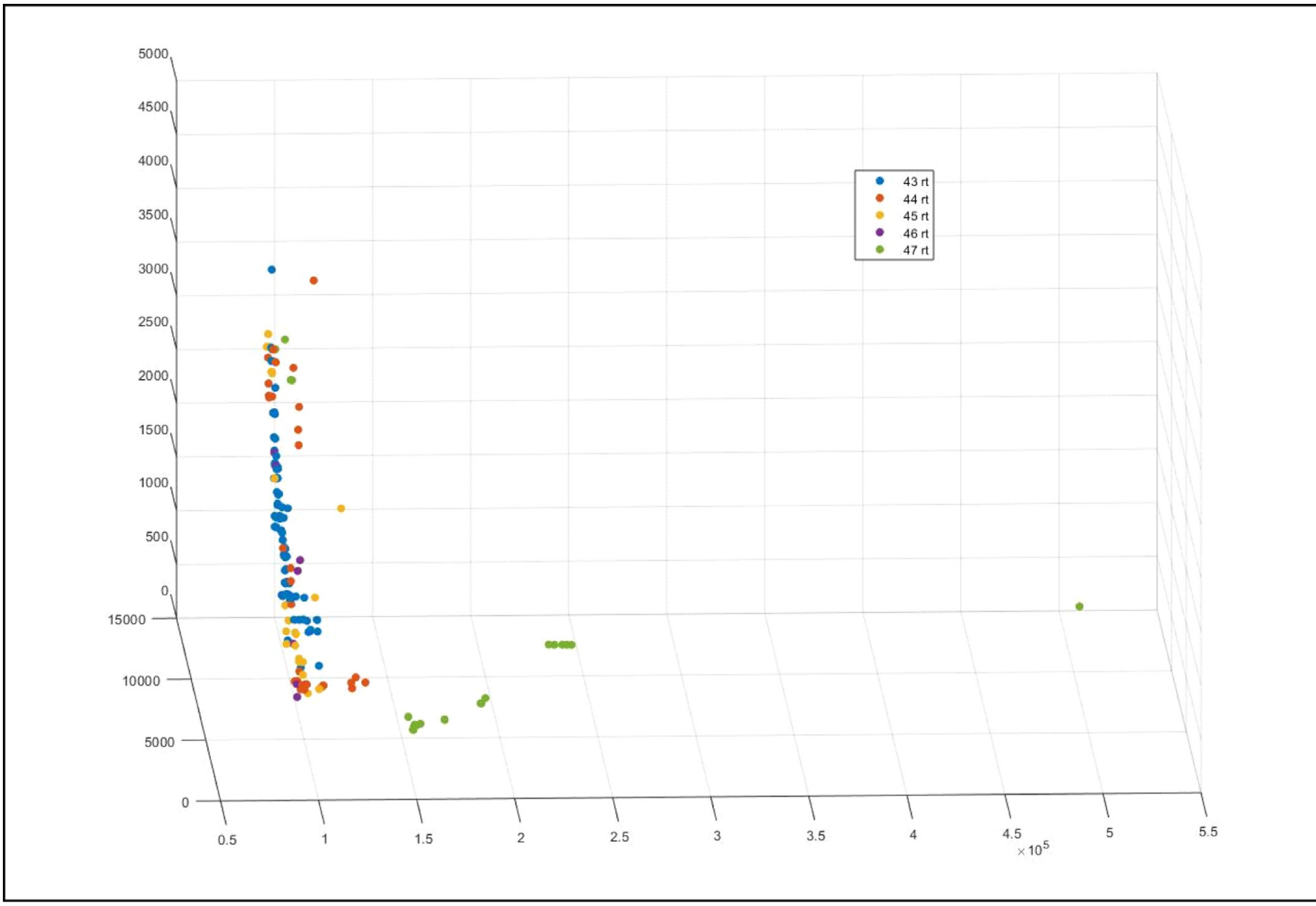

Fig. 6. Approximations of the Pareto front of IG_VNS for different f_4-values.

to the definition of the objectives considered, we would like to recall the particularities of the problem, for a better understanding of the results (as described in Section 3). In particular, it is important to take into account that we are proposing a daily collection, instead of studying the frequency, in order to improve customers' service.

Hence, we conclude our study showing in Fig. 6 the frontiers obtained with our method on the real waste collection problem shown in Fig. 1 with 991 collecting points. In particular, as show in Fig. 6, we consider 5 different values for the number of routes, starting from $f_4 = 7$ (objective $f_4 = 7,\ 8,\ 9,\ 10,\ 11$). This figure shows the typical front in combinatorial optimization problems; i.e., due to the some constraints, such as worker's shifts and capacity (in our case), the approximation of the Pareto front presents some discontinuities with some isolated points (also, non-dominated solutions).

An interesting exercise is to compare our solutions with the one currently implemented by the local government. It is difficult in mathematical terms because we are obtaining many points in our approximations, while the current solution is just one single point. Additionally, in practical terms their approach covers the service with a periodical routing system and we improve clients' satisfaction by running a daily collection service. Anyway, to have a first indicator, we can compare the current solution with some of our solutions. In particular, Table 8 shows the solutions that we obtained when optimizing each objective independently (the so-called payoff matrix).

Table 8 shows in the first row the solution currently implemented, and summarizes the solutions obtained when optimizing f_1, f_2 and f_3, in rows 2, 3, and 4 respectively. Note that the value of f_4 ranges between 7 and 11 in these solutions. It is clear that the objective values of the current solution (row 1) are relatively far from the best ones (depicted in bold). In particular, $f_1 = 847,304.00$, which represents a percentage deviation of 115.02% with respect to the best value that we obtained of 394,044.99. This clearly indicates that the solution of our method will provide the user with significantly shorter routes, with a total sum of distances much lower than the existing ones. A similar pattern is observed for objective function f_2 where the current solution is on a 375.45% of the best one obtained. This is especially relevant in terms of the practical implications of our best solution for f_3, which turns out to be much more balanced than the existing one (the best value for this objective is 352.54, which is significantly lower than the 26,004.60 currently implemented). As a matter of fact, we already received a very positive feedback from the planning team providing this service, when discussing the new solutions and associated route plan. The current solution is based on 7 routes, while we consider many more possibilities (from 7 to 11) when solving the problem. It must be noted however, that some of our best solutions, as the best one for f_1 and the best one for f_3 also have 7 routes, indicating that the experience of the team somehow pro-

Table 8
Current solution and payoff matrix.

Solution	f_1	f_2	f_3	f_4
Currently implemented	847304.00	237543.00	26004.60	7
Best one for f_1	**394044.99**	86997.69	24754.47	7
Best one for f_2	588552.12	**49961.94**	23137.00	11
Best one for f_3	1175883.70	229819.00	**352.54**	7

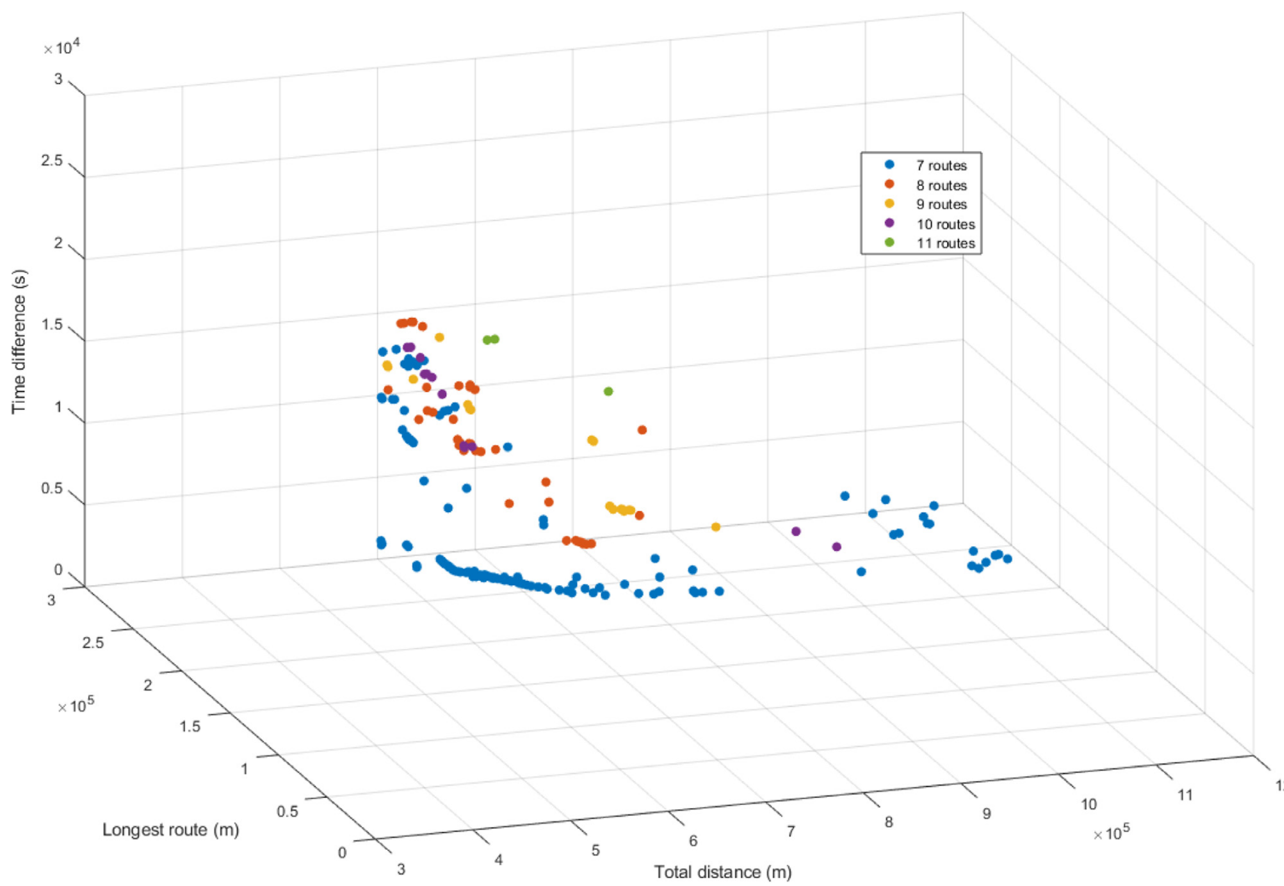

Fig. 6. Approximations of the Pareto front on the real instance shown in Fig. 1.

duced a solution compatible with most of our solutions, although with different configurations. This was received in a positive way by the planning team, and helped us to explain them, starting with a solution with the same number of routes, how alternative solutions (in particular those that we obtained in the approximation of the efficient frontier) can provide a better service from different perspectives. We are analyzing now with them these alternatives. To sum it up, considering that we provide the decision maker with many options (non-dominated solutions) with very good values on the objectives considered (as compared to their current ones), we can conclude that, in mathematical terms, our solution method solves satisfactorily this real problem, and in practical terms, it receives a positive feedback from the company.

8. Conclusions

In this paper, we investigate the adaptation of the Iterated Greedy and the Variable Neighborhood Search (VNS) methodologies to a multiobjective routing problem. This study was motivated by a real waste collection problem and the need of providing solutions to the local government. We extend its scope and undertake a scientific study on search strategies to this optimization problem, including public domain instances to draw significant conclusions. In particular, we focus on the effect of the balance between randomization and greediness on the performance of our heuristic methods when solving this NP-hard problem.

From a methodological point of view, Iterated Greedy is a powerful metaheuristic based on constructing and partially destructing solutions to reconstruct them iteratively. The application of constructive neighborhoods has its antecedents in Strategic Oscillation, which has proved very effective of guiding search methods to explore the solution space economically and effectively. Most of these antecedents however have been tested in mono objective problems. In this work, we propose to extend this methodology to the multi-objective case by means of the Wierzbicki achievement scalarizing function. This research confirms that the combination between this metaheuristic coupled with VNS and the guiding function is able to obtain high-quality solutions. After exhaustive experimentation, we can conclude that our method obtains solutions of better quality (according to the standard multiobjective indicators) than those obtained with NSGA-II.

The algorithm presented in this work is designed to help the expert in the company (administration in this case) deciding the routes followed by waste collection trucks. Currently, those routes are evaluated manually, including involuntary human mistakes that are really difficult to correct. The inclusion of the proposed algorithm in the expert system will allow the local administration to better estimate the necessities in terms of vehicle and personnel, leading them to save both time and money in the waste collection of Antequera (Spain).

Declaration of Competing Interest

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

Acknowledgement

The authors would like to thank the anonymous referees for their comments and suggestions to improve the paper. This work has been partially supported by the Spanish Ministerio de

Ciencia, Innovación y Universidades (MCIU/AEI/FEDER, UE) with grant refs. PGC2018-095322-B-C21, ECO2016-76567-C4-4-R and PGC2018-095322-B-C22; and Comunidad de Madrid y Fondos Estructurales de la Unión Europea with grant ref. P2018/TCS-4566.

References

Alumur, S., & Kara, B. (2007). A new model on hazardous waste location-routing problem. *Computers & Operations Research, 34*(5), 1406–1423.

Angelelli, E., & Speranza, M. (2002). The application of a vehicle routing model. *Journal of the Operational Research, 53*(9), 944–952.

Arribas, C., Andrea Blazquez, C., & Alejandra Lamas, A. (2010). Urban solid waste collection system using mathematical modelling and tools of geographic information systems. *Waste Management and Research, 28*, 355–363.

Baptista, S., Oliveira, R. C., & Zúquete, E. (2002). A period vehicle routing case study. *European Journal of Operational Research, 139*(2), 220–229.

Bautista, J., Fernández, E., & Pereira, J. (2008). Solving an urban waste collection problem using ants heuristics. *Computer Operations Research, 35*, 3020–3033.

Beliën, J., De Boeck, L., & Van Ackere, J. (2012). Municipal solid waste collection and management problems: A literature review. *Transportation Science, 48*(1), 1–37.

Benjamin, A., & Beasley, J. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research, 37*, 2270–2280.

Benjamin, A., & Beasley, J. (2013). Metaheuristics with disposal facility positioning for the waste collection VRP with time windows. *Optimization Letters, 7*, 1433–1449.

Bing, X., Bloemhof, J., Rodrigues Pereira Ramos, T., Barbosa-Povoa, A., Wong, C., & van der Vorst, J. (2016). Research challenges in municipal solid waste logistics management. *Waste Management, 48*, 584–592.

Caballero, R., Laguna, M., Martí, R., & Molina, J. (2011). Scatter Tabu search for multiobjective clustering problems. *Journal of the Operational Research Society, 62*, 2034–2046.

Cattaruzza, D., Nabil, A., Dominique, F., & González-Feliu, J. (2017). Vehicle routing problems for city logistics. *European Journal of Transportation and Logistics, 6*, 51–79. doi:10.1007/s13676-014-0074-0.

Chalkias, C., & Lasaridi, K. (2009). A gis based model for the optimisation of municipal solid waste collection: The case study of Nikea, Athens, Greece. *WSEAS Transactions on Environment and Development, 5*, 640–650.

Chang, N.-. B., & Wei, Y. (1999). Strategic planning of recycling drop-off stations and collection network by multiobjective programming. *Environmental Management, 24*, 247–263.

Constantino, M., Gouveia, L., Ao, M. M., & Nunes, A. (2015). The mixed capacitated arc routing problem with non-overlapping routes. *European journal of Operational Research, 244*(2), 445–456.

Corberán, A., & Laporte, G. (2015). Routing in waste collection. In Arc routing: Problems, methods and applications (pp. 351–371). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Cortinhal, M. J., Mourao, M. C., & Nunes, A. C. (2016). Local search heuristics for sectoring routing in a household waste collection context. *European Journal of Operational Research, 255*, 68–79.

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

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Durillo, J., & Nebro, A. (2011). jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software, 42*, 760–771.

Erfani, S., Danesh, S., Karrabi, S., & Shad, R. (2017). A novel approach to find and optimize bin locations and collection routes using a geographic information system. *Waste Management and Research, 35*(7), 776–785.

Faccio, M., Persona, A., & Zanin, G. (2011). Waste collection multi objective model with real time traceability data. *Waste Management, 31*(12), 2391–2405.

Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*, 109–134.

Ferreira, J. A., Costa, M., Tereso, A., & Oliveira, J. A. (2015). A multi-criteria decision support system for a routing problem in waste collection. In *Proceedings of the 8th international conference on evolutionary multi-criterion optimization, EMO*. Springer.

Fisher, M. L., & Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks, 11*, 109–124.

García-Martinez, Glover, Rodriguez, Lozano, & Martí (2014). Strategic oscillation for the quadratic multiple knapsack problem. *Computational Optimization and Applications, 58*(1), 161–185.

Garcia-Najera, A., & Bullinaria, J. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 38*(1), 287–300.

Ghose, M., Dikshit, A., & Sharma, S. (2006). A GIS based transportation model for solid waste disposal - A case study on Asansol municipality. *Waste Management, 26*(11), 1287–1293.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boulder, Colorado: Kluwer Academic Publishers Norwell.

Gómez, J., Pacheco, J., & Gonzalo-Orden, H. (2015). A tabu search method for a bi-objective urban waste collection problem. *Computer Aided-Civil and Infraestructure, 30*, 36–53.

Guerrero, L., Mass, G., & Hogland, W. (2013). Solid waste management. *Waste Management, 33*(1), 220–232.

Haimes, Y., Lasdon, L., & Wismer, D. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-1*(3), 296–297.

Han, H., & Ponce-cueto, E. (2015). Waste collection vehicle routing problem: Literature review. *Traffic Engineering, 27*, 345–358.

Hanafi, S., Freville, A., & Vaca, P. (1999). Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR: Information Systems and Operational Research, 37*(3), 236–254.

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization, 5*(3), 423–454.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics, 19*, 129–156.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Vigo, D. (2014). Models and algorithms for the integrated planning of bin allocation and vehicle routing in solid waste management. *Transportation Science, 48*, 103–120.

Herrán, A., Colmenar, J. M., Martí, R., & Duarte, A. (2020). A parallel variable neighborhood search approach for the obnoxious p-median p-dispersion problem. *International Transactions in Operations Research, 27*, 336–360.

Jacobsen, S., & Madsen, O. (1980). A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research, 5*, 378–387.

Jozefowiez, N., Semet, F., & Talbi, E. G. (2007). Multi-Objective vehicle routing problems. *European Journal of Operational Research, 189*, 293–309.

Jozefowiez, N., Semet, F., & Talbi, E.-. G. (2009). An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research, 195*(3), 761–769.

Karadimas, N., Papatzelou, K., & Loumos, V. (2007). Optimal solid waste collection routes identified by the Ant Colony System algorithm. *Waste Management and Research, 25*(2), 139–147.

Kim, B.-. I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research, 33*, 3624–3642.

Li, R., Leung, Y., Lin, H., & Huang, B. (2013b). An adaptive compromise programming method for multi-objective path optimization. *Journal of Geographical Systems, 15*, 211–228.

Li, R., Leung, Y., Lin, H., & Huang, B. (2013a). A genetic algorithm for multiobjective dangerous goods route planning. *International Journal of Geographical Information Science, 27*(6), 1073–1089.

Li, X., & He, J. (2009). Intelligent gis for solving high dimensional site selection problems using Ant Colony System technology. *International Journal of Geographical Information Science, 23*(4), 399–416.

Liao, C., & Lin, C. (2003). Makespan minimization for two uniform parallel machines. *International Journal of Production Economics, 84*, 205–213.

López-Sánchez, A., Hernández-Díaz, A., & Gortázar, F. &. (2017). A multiobjective GRASP - VND algorithm to solve the waste collection problem. *International Transactions in Operational Research, 25*(2), 545–567.

Maniezzo, V., & Roffilli, M. (2008). Algorithms for large directed capacitated arc routing problem instances. In C. Cotta, & J. van Hemert (Eds.), *Recent Advances in Evolutionary Computation for Combinatorial Optimization (Vol. 153)* (pp. 259–274). Springer-Verlag: Berlin, Heidelberg.

Marks, D., & Liebman, J. (1970). *Mathematical analysis of solid waste collection: final Final report*. Public Health Service publication: U.S. Government Printing Office.

Marshall, R. E., & Farahbakhsh, K. (2013). Systems approaches to integrated solid waste management in developing countries. *Waste Management, 33*(4), 988–1003.

Martí, R., Campos, V., Resende, M., & Duarte, A. (2015). Multi-objective GRASP with path-relinking. *European Journal of Operational Research, 240*, 54–71.

Miettinen, L. (1999). *Nonlinear multiobjective optimization*. Boston: Kluwer Academic Publishers.

Mladenovic, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*(11), 1097–1100.

Mole, R., & Jameson, S. (1976). A sequential route-building algorithm employing a generalized saving criterion. *Operational Research Quarterly, 27*, 503–511.

Nguyen-Trong, K., Nguyen-Thi-Ngoc, T., Nguyen-Ngoc, D., & Dinh-Thi-Hai, V. (2017). Optimization of municipal solid waste transportation by integrating GIS analysis, equation based and agent-based model. *Waste Management, 59*, 14–22.

Ombuki, B., Ross, B., & Hanshar, F. (2006). Multi-Objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence, 24*, 17–30.

Oyola, J., & Løkketangen, A. (2014). GRASP-ASP: An algorithm for the CVRP with route balancing. *Journal of heuristics, 20*(4), 361–382.

Pacheco, J., & Martí, R. (2006). Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society, 57*(1), 29–37.

Pardo, E. G., Mladenovic, N., Pantrigo, J., & Duarte, A. (2013). Variable formulation search for the Cutwidth Minimization Problem. *Applied Soft Computing, 13*, 2242–2252.

Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *The Journal of the Operational Research Society, 46*(12), 1433–1446.

Ruíz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research, 177*(3), 2033–2049.

Samanlioglu, F. (2013). A multi-objective mathematical model for the industrial hazardous waste location-routing problem. *European Journal of Operational Research, 226*(2), 332–340.

Sánchez-Oro, J., Laguna, M., Martí, R., & Duarte, A. (2016). Scatter search for the Bandpass problem. *Journal of Global Optimization, 66*(4), 769–790.

Sánchez-Oro, J., Martínez-Gavara, A., Laguna, M., Martí, R., & Duarte, A. (2017). Variable neighborhood scatter search for the incremental graph drawing problem. *Computational Optimization and Applications, 68*, 775–797.

Simonetto, E., & Borenstein, D. (2007). A decision support system for the operational planning of solid waste collection. *Waste Management, 27*, 1286–1297.

Stam, A., Kuula, M., & Cesar, H. (1992). Transboundary air pollution in Europe: An interactive multicriteria trade-off analysis. *European Journal of Operational Research, 56*, 263–277.

Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., & (2008). Multiobjective optimization: Interactive and evolutionary approaches. In *Real-World applications of multiobjective optimization* (pp. 285–327). Heidelberg: Springer Berlin.

Tavares, G., Zsigraiova, Z., Semiao, V., & Carvalho, M. (2009). Optimization of WSW collection routes for minimum fuel consumption using 3D GIS modeling. *Waste Management, 29*, 1176–1185.

Tung, D. V., & Pinnoi, A. (2000). Vehicle routing - scheduling for waste collection in HANOI. *European Journal of Operational Research, 125*, 449–468.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., & Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research, 257*(3), 845–858.

Wierzbicki, A. (1980). The use of reference objective in multiobjective optimization. In *Multiple criteria decision making theory and application* (pp. 468–486). Heidelberg: Berlin-Springer.

Xue, W., & Cao, K. (2016). Optimal routing for waste collection: A case study in Singapore. *International Journal of Geographical Information Science, 30*(3), 554–572.

Ying, K., & Cheng, H. (2010). Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications, 37*(4), 2848–2852.

Zitzler, E. L. (2003). SPEA2: Improving the strength pareto evolutionary algorithm. *Evolutionary methods for design, optimization and control with applications to industrial problems*. Athens, Greece.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: methods Methods and applications* Ph.D. Thesis. Zurich: Swiss Federal Institute of Technology.

Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative case. In *Proceedings of the 5th international conference on parallel problem solving from nature* (pp. 292–301). Springer.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & Da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computations, 7*(2), 117–132.