

Research Article

An Interactive Biobjective Method for Solving a Waste Collection Problem

L. Delgado-Antequera,¹ F. Pérez,² A. G. Hernández-Díaz,³ and A. D. López-Sánchez³

¹Programa de Doctorado de Economía y Empresa de la Universidad de Málaga, Campus El Ejido s/n, 29071 Málaga, Spain

²Universidad de Málaga, Campus El Ejido s/n, 29071 Málaga, Spain

³Pablo de Olavide University, Carretera de Utrera Km 1, 41013 Seville, Spain

Correspondence should be addressed to L. Delgado-Antequera; lauda1g10@uma.es

Received 21 January 2016; Accepted 29 March 2016

Academic Editor: Danielle Morais

Copyright © 2016 L. Delgado-Antequera et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The aim of this paper is to propose a framework in order to solve the real-world waste collection problem in a city of southern Spain modeled as an Asymmetric Vehicle Routing Problem (AVRP) with side constraints and several variations. In this problem, not only are vehicle capacity and temporal constraints considered but multiple trips are also allowed. Furthermore, two objectives will be considered: the minimization of the total distance and the balance of the working day. Finally, in order to select a single solution among all efficient (or nondominated) solutions, an interactive method is designed using reference points.

1. Introduction

In the public sector, lots of benefits are derived from a good decision-making process. This is why in the last few years the authorities have devoted an increasing attention on this topic to solve this kind of problems in the most satisfying and efficient way. In our particular case, a real-world waste collection problem (WCP) is proposed, where labour, economic, social, and environmental aspects are considered. This essential service involves large operational costs, and so researchers strive to reduce these costs through improving the routing of waste collection, as well as the determining the most suitable location of disposal facilities (or containers) or minimizing the number of vehicles employed. The complexity of this situation increases in the presence of multiple objectives that must be satisfied simultaneously, when satisfying the particular constraints of the problem.

An early paper on waste collection was published by [1] and focused on waste collection activities in New York City (USA). They were interested in the design of routes in accordance with feasible combinations of days for the collection of containers for exactly a preset number of times. The objective was to service all containers assigned each day

and to minimize the overall routing cost. In particular, they explored a variety of routing procedures and addressed the problem in two different ways: they clustered first and then optimized each of the routes; and they routed first and then partitioned this giant tour into feasible routes. Numerous researchers have since then studied similar models and have developed a variety of methods to solve the WCP so the evolution of the number of studies on waste has greatly increased over the years. Recently, other approaches have been published in this field. This is the case of [2], [3], or [4], in which a multiobjective tabu search is proposed for urban waste collection problems.

In this paper, we will solve the biobjective real-world waste collection problem in a city of southern Spain modeled as an Asymmetric Vehicle Routing Problem (AVRP) with side constraints and several variations. In the considered area, the waste is daily collected. The network road of the city can be modeled as a directed graph $G = (V, A)$, where V is the set of nodes and A is the set of arcs. Each node $i \in V$ represents a container and each arc $(i, j) \in A$ is the shortest path between two containers $(i, j \in V)$. One-way streets are represented with an arc and two-way streets are represented with two different arcs (each representing one of the two

possible directions). Additionally, we consider a special node called the *depot* (which in fact is also the *landfill*) and denoted as node 0, where vehicles are parked and emptied.

In this problem, in contrast to most of the models described in the literature, various types of costs are considered: the *travel cost*, c_{ij} , that is proportional to the distance (or length) of the corresponding arc $(i, j) \in A$; the *service cost*, s_{ij} , associated to each arc, $(i, j) \in A$, that represents the time needed to go from container $i \in V$ to container $j \in V$; and the *dumping cost*, λ , that is the time spent by a vehicle when it is dumping at the landfill. Each container $i \in V$ has associated a quantity, q_i , of waste that must pick it up. Thus, two conflicting objectives are considered in this paper: (1) minimizing the total distance travelled by all vehicles and (2) minimizing the length of the longest route in order to balance the working day within the fleet of vehicles. Regarding the constraints, the total time cost of each vehicle is bounded by the working day. Also, the quantity of waste collected must be considered, since our problem is capacity constrained by the tare of the truck.

A solution of this problem consists of designing K routes where each route must start and finish at the depot and the time spent performing each route cannot exceed the available working time, W . Each vehicle, with a limited capacity Q , must perform a *route* that will be defined as a sequence of T trips. Therefore, a *vehicle's route* is defined as the total course driven per day from departing from the depot, until returning back to the depot.

This problem is related to the well-known family of the Asymmetric Capacitated Vehicle Routing Problem (ACVRP); see [5] or [6]. Although our problem and the ACVRP are connected, the real-situation proposed makes us include other constraints. For instance, the classical ACVRP constructs only one simple trip while we allow the construction of routes with multiple trips; another variation comes since ACVRP imposes only capacity constraints and we also need to include time constraints. Finally, the ACVRP considers just one objective, the minimization of the total distance. In contrast, we consider also the minimization of the longest route.

Given all the above, an interactive algorithm for a biobjective multistart algorithm based on reference point is implemented. In general, interactive multiobjective methods assume that decision-maker (DM) is able to provide consistent feedback regarding the preferences to be included in the resolution process in order to guide the search towards certain areas of the Pareto front. To accomplish a set of efficient solutions a multistart algorithm is implemented using the ε -constraint method and then the interactive phase is used to guide the search and select the most preferable solutions by the DM.

As we can see in the literature, many different heuristic and metaheuristic strategies (see [7–9]) have been developed and applied in the resolution of problems with multiple objectives and several constraints. These methodologies are mainly defined by the search method (local search, scatter search, tabu search, and so on) used to create the initial set of efficient solutions. However, not many papers apply interactive methods in order to obtain solution of

multiobjective routing problems. Furthermore, a reduced number of these publications are directly related to the waste collection problem (see [10–12]), where the authors design a tool which allows the user to manipulate the database in the routes-construction phase. Visual alternative is presented to the managers using a Geographic Information System (GIS) which helps to the decision-maker to select the closest solution to its requirements.

The rest of the paper is organized as follows. Section 2 presents the proposed algorithm for the solution of the considered problem, and Section 3 provides the computational results applied on the real-world WCP described. Finally, Section 4 summarizes the paper and discusses future work.

2. The Proposed Algorithm

In this section, a competitive multistart algorithm is proposed to solve the considered problem. This algorithm is an iterative process, in which each iteration consists of two phases: the first one (called *construction phase*) in which a feasible solution is generated and the second one (called *local search phase*) in which it tries to improve the solution quality by using a set of neighbourhoods. As a consequence, each iteration produces a locally optimal solution. The best overall solution is selected as a result of the whole procedure.

The proposed multistart algorithm has the following characteristics. It is a Memory-less, Randomized, and Build-from-scratch algorithm. According to [13], Memory-less means that there are no elements that are common to certain previously generated solutions, Randomized means that the starting solutions are randomly generated, and Build-from-scratch means that none of the elements remain fixed from one generation to another.

As we are solving a biobjective problem, the well-known ε -constraint method is used. The traditional ε -constraint method optimizes one of the objectives, when the others are introduced into the constraint space for guaranteeing that basic requirements are satisfied (see [14], [15], or [16]). This method is appropriate for a combinatorial problem where non-big-size Pareto front is expected and suitable for evolutionary algorithms (see [17]). Also, the cost of running multiple executions for the related single-objective problem treated is not elevated, since the algorithm proposed is significantly fast although a multiobjective extension of the methodology will be investigated in the future introducing specific constructive and local searches for each of the objectives. Other improved ε -constraint methods especially suitable for Multiobjective Integer Programming (MOIP) problems have been proposed in [18–20].

With regard to the problem treated, the minimization of the distance is included as the primary objective and the minimization of the longest route is included in the form of an inequality constraint in order to accomplish a set of efficient solutions.

Once the set of efficient solutions is obtained an interactive approach is designed. We focus on interactive approaches because they are effective methods to deal with multiobjective problems without having an excessive computational effort. We use an interactive scheme which gradually leads to the

areas of the efficient set that are more attractive to the DM. Thus, on one hand, computational effort can be reduced and on the other hand, the DM is assisted in choosing the best solution according to his/her preferences.

2.1. Construction Phase. In the construction phase a feasible solution must be obtained. We have designed a multistart algorithm to solve the WCP in which a new algorithm has been employed to construct the initial routes. The algorithm is based on the Mole and Jameson insertion heuristic (see [21]), known as the *extra cost* combined with a more elaborate algorithm, called the *regret* heuristic (see [22]).

Recall that the *extra cost function*, $f^{(1)}$, computes the change in the objective function when inserting each unserved container, $j \in J$, at the best position, i , in route r_k . In mathematical terms, $f^{(1)}(j, r_k, i) = c_{r_k(i-1)j} + c_{jr_k(i)} - c_{r_k(i-1)r_k(i)}$, where $r_k(i)$ denotes the node in the i th position on route r_k . The algorithm selects the node with the lowest value of the extra cost function; that is, the extra cost function should be minimized. This strategy adds containers to routes while the capacity of a vehicle or the working-time constraint is not exceeded. However, the *regret function*, $f^{(2)}$, computes the difference in cost when inserting each unserved container, j , into the best route and when inserting it into the second-best route (obviously in its best position). More formally, $f^{(2)}(j, r_k, i) = f^{(1)}(j, r_{k_2}, i_{k_2}) - f^{(1)}(j, r_k, i)$, where r_k denotes the route with the lowest extra cost and r_{k_2} denotes the route with the second-lowest extra cost. In contrast with the extra cost function, the regret function should be maximized.

Even if the extra cost function is easier to implement and faster than the regret function, a problem related to the extra cost function is that it often postpones the insertion of the most difficult nodes (relatively large values of $f^{(1)}$) until the last iterations. In this regard, the algorithm retains very few alternatives for their insertion. The regret function strives to circumvent the problem by incorporating a kind of forecasting information when selecting the node to be inserted.

In this paper, the extra cost function and the regret function are mixed in order to keep the best features of these both functions. The new proposed algorithm will be called *mixture of regret and extra cost*. This heuristic inserts the first $\alpha\%$ nodes using the regret function and the last $(1-\alpha)\%$ nodes using the extra cost function, where $\alpha \in [0, 1]$.

It is needed to emphasize that, as first step of the algorithm, all routes are initialized by including one of the most difficult nodes to service. That is, we select the K nodes furthest away from the depot and from each other. That is, in order to select the most difficult containers to serve, the next function, $g^{(0)}$, is considered, which computes the distance between unserved containers, J , and served containers. In mathematical terms, given the set of served containers, S , this function is computed for each unserved container, j , as $g^{(0)}(j) = \sum_{s \in S \cup \{0\}} \min\{c_{sj}, c_{js}\}$. A deterministic strategy would select the node with the maximum value of $g^{(0)}$. Instead, this approach constructs a list with a percentage β of the best candidates (according to the considered function) and then a node of the list is selected at random and all the involved

variables are updated. The process is maintained until the K routes are initialized with only one node. Once all routes are initialized, the remaining unserved nodes are included using the function $g^{(1)}$ (i.e., minimizing $f^{(1)}$ or $-f^{(2)}$ according to the considered inclusion strategy).

Algorithm 1 shows the pseudocode of the constructive procedure. As input parameters, the algorithm receives the graph, G , the number of vehicles, K , β , the parameter of the initial nodes, and the parameter of the mixture α which control the balance of the randomness (line (1)) and returns a feasible set of routes (line (2)). The algorithm starts by initializing the set of routes with the highest cost, that is, the most difficult containers to serve (lines (4) to (9)).

The proposed multistart construction fails to guarantee the feasibility of the obtained solution (although these details have not been introduced into the pseudocode for the sake of simplicity). Specifically, due to the randomness in the construction of the initial solution, it could happen that one or a few nodes cannot be inserted at any of the already built routes, because of the capacity or time constraint. Then some nodes might remain unvisited, so one of the basic statements of the vehicle routing problem is not satisfied. Thus, when the solution is not feasible, it is discarded and the procedure triggers a new iteration to construct a new solution. Only feasible solutions are submitted to the local search phase.

2.2. Local Search Phase. Due to the size of the optimization problem under consideration, relatively fast procedures are needed in order to attain a solution within a reasonable computing time. However, potential solutions to the local search phase also need to be provided. To this end, the aforementioned constructive procedure builds very diverse solutions. Therefore, the local search procedure will explore the solution space starting from very different solutions.

The WCP presents a search space with an enormous quantity of local optima. Therefore, traditional local search methods (based only on one neighbourhood) fails to perform well on this kind of problem. For this reason, we propose a Variable Neighbourhood Descent (VND) to overcome these difficulties; see [23] or [24]. In VND, several different neighbourhoods are explored, in this case from the largest and slowest to evaluate to the smallest and fastest. The process iterates over each neighbourhood while the solution is improved, performing local searches until a local optimum is found in each neighbourhood. Only strictly better solutions are accepted after each neighbourhood search. In VND, the returned solution is a local optimum in each neighbourhood, \mathcal{N}_m , with $1 \leq m \leq m_{\max}$ different neighbourhoods. Therefore, the global optimum is likely to be found earlier than when considering only one neighbourhood.

The pseudocode of the local search is shown in Algorithm 2 where a nested strategy is considered. It has only two input arguments: an initial solution r (line (2)) and the number of neighbourhoods, m_{\max} . Our VND algorithm uses three neighbourhood structures.

- (1) Neighbourhood structure \mathcal{N}_1 : a chain of λ consecutive nodes from the route r_i are included in route r_j , where $i \neq j$, and a chain of μ consecutive

```

(1) Input:  $G = (V, A), K, \beta, \alpha$ ;
(2) Output:  $r = (r_k)_{k=1, \dots, K}$  feasible set of routes;
(3) Initialize:  $S = \{0\}, J = V \setminus \{0\}, k \leftarrow 1, r \leftarrow \emptyset, f^* \leftarrow \infty$ ;
(4) repeat
(5)   For all  $j \in J$  compute  $g^{(0)}(j)$ ;
(6)   Define the list of seeds:  $\{j \in J \mid g^{(0)}(j) \geq g_{\max}^{(0)} - \beta(g_{\max}^{(0)} - g_{\min}^{(0)})\}$ ;
(7)   Select  $j^*$  at random from the list of seeds;
(8)    $r_k \leftarrow r_k \cup \{j^*\}, S \leftarrow S \cup \{j^*\}, J \leftarrow J \setminus \{j^*\}, k \leftarrow k + 1$ ;
(9) until  $k = K + 1$ ;
(10)  $r \leftarrow r, J \leftarrow J$ ;
(11) while  $J \neq \emptyset$  do
(12)   while the capacity or the working day of the  $K$  vehicles is not exceeded do
(13)     For all  $j \in J$  compute  $g^{(1)}(j, k, i)$ ;
(14)     Select  $j^*$  such as  $g^{(1)}(j^*, k, i) < g^{(1)}(j, k, i)$ ;
(15)      $r_k \leftarrow r_k \cup \{j^*\}, J \leftarrow J \setminus \{j^*\}$ ;
(16)   end while
(17) end while
(18) return  $r$ 

```

ALGORITHM 1: Construction phase.

```

(1) Define the set of neighbourhood structures  $\mathcal{N}_m, m = 1, \dots, m_{\max}$ 
(2) Input:  $r$  feasible solution;
(3) Initialize:  $m \leftarrow 1$ ;
(4) repeat
(5)   local search:  $r' \leftarrow \arg \min \mathcal{N}_m(r)$ ;
(6)   if  $f(r') < f(r)$  then
(7)      $r \leftarrow r'$  and  $m \leftarrow 1$ ;
(8)   else
(9)      $m \leftarrow m + 1$ ;
(10)  end if
(11) until  $m = m_{\max} + 1$ ;
(12) return  $r$ 

```

ALGORITHM 2: VND.

nodes from the route r_j are included in route r_i . This is an interroute movement that selects all pairs of possible routes and checks which nodes can be swapped between the routes. This is repeated until all pairs of routes have been considered and the best movement is performed.

- (2) Neighbourhood structure \mathcal{N}_2 : a node is moved from one route to another, that is, this is an interroute operator. This exchange evaluates the possibility of moving one node belonging to a route r_i in the best position of any other route r_j , where $i \neq j$, in an attempt to improve the total distance. We select one route and we check which nodes can be inserted from another route. This is repeated until all possibilities have been considered and the best movement is performed.
- (3) Neighbourhood structure \mathcal{N}_3 : two consecutive nodes are exchanged within a route, that is, this is an intraroute movement in order to reduce the current route distance.

The procedure starts by obtaining a local optimum r' with respect to the first neighbourhood. Instead of abandoning the search (as a local search procedure), VND then resorts to the following neighbourhood searching for an improvement. If an improvement is found, the search starts again by considering the first neighbourhood (which implies setting $m = 1$). Otherwise, VND explores the next neighbourhood by increasing m (until m_{\max} is reached).

2.3. Interactive Phase. The DM's preferences have to be included to select a solution within that set and few studies have addressed this issue in a metaheuristic context. We use an interactive procedure based on reference points although many other valid methods can be used such as the utility function method, lexicographic method, goal programming, weighting methods, and ε -constraint (see the book [25] for an excellent review about this topic). Reference-point-based methods are one of the most natural ways of expressing preference information in solutions. It eases the process to the DM, who wants to reach them for the objective functions. Then, the primary aim of the method is to reduce

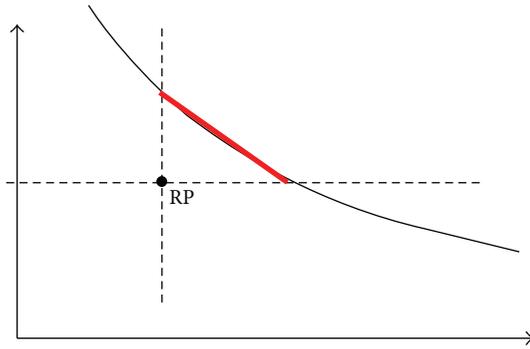


FIGURE 1: Unfeasible reference point.

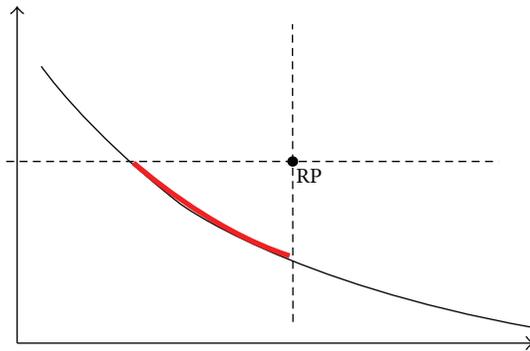


FIGURE 2: Feasible reference point.

the size of the approximation of the efficient frontier, using the information iteratively supplied by the DM. Then, the DM must provide the desired values for each objective (i.e., a reference point RP). Thus, the main idea is to project this point on the efficient frontier to obtain the efficient point closer to this reference point (converting the original MO problem into a single-objective optimization problem). However, our method provides a set of efficient solutions in the area where this projected solution lies, rather than a single efficient solution. When the aim is to minimize two objectives, this approach (using the reference point) will define the area of the efficient frontier where the interesting solutions are expected to be. Also, an interesting advantage comes from the fact that it is not a matter of fact that the reference point is feasible or not. This is illustrated in Figures 1 and 2 where the whole frontier is reduced to the highlighted portion. The iterative process continues until the DM reaches the best compromise regarding the best solution according to his/her preferences.

The proposed interactive process is based on the g -dominance concept proposed by [26], but, in our case, the efficient frontier obtained is not modified along the interactive phase. The g -dominance method is easy to include to any multiobjective metaheuristic strategy used, since it does not imply the modification of the main architecture of the specific search engine adopted. Apart from this, it might be applied a posteriori, to filter an area of the Pareto front, or included in the algorithm looking for its convergence to the desired zone. Additionally, this method allows the modification of the

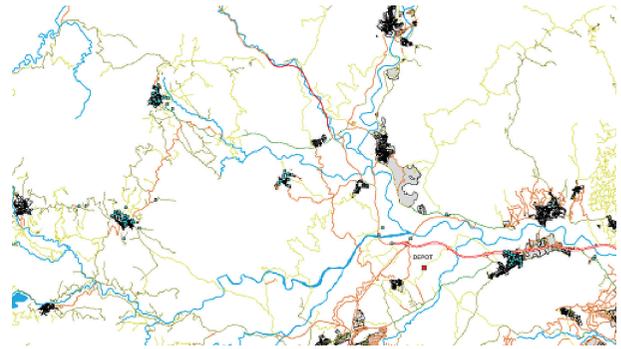


FIGURE 3: Localization of containers.

reference point during the searching process. On the other hand, one of its main disadvantage is that any of the reference point methods requires the knowledge of the ideal and anti-ideal point, which is not always easy to obtain.

As shown above, at each iteration, a representative subset of efficient routes, according to the offered reference point RP, is shown to the DM. If the DM does not feel satisfied with any of these solutions, he/she can modify the reference point in order to refine his/her preferences.

3. Computational Results

In this section we show the computational results when solving the WCP by using the proposed iterative algorithm. The code has been implemented using the programming language C# in Visual Studio 2010. The program was run in a Samsung Series 5 NP535U3C notebook; Windows 7 Home Premium (64-bit); CPU: AMD Dual-Core A6-4455M Accelerated Processor; speed (GHz): 2.1 GHz; CPU Cache: 1 MB; and System Memory: 4 GB.

This problem deals with four trucks which are used to satisfy the collection of 214 rear-loading containers daily. The containers are located in an area at the south of Spain and are represented in Figure 3 by the use of a GIS.

This problem belongs to the set of Capacitated Vehicle Routing Problems (CVRP), since each truck is limited to collect up to 10,220 kg. Moreover, an additional constraint on the duration of the routes, of 37.5 hours per week, has been taking into consideration, in order to satisfy the work-day requirements of the company in charge. And the following objective functions are considered:

- (1) f_1 : minimize the total distance driven by all of the vehicles.
- (2) f_2 : minimize the longest route performed within the set of vehicles.

The complete algorithm was run taking into account the two objective functions previously defined and the respective constraints of the problem. Then, a set of 31 efficient Pareto points were identified. Table 1 contains a sample of these solutions, where Sol. 1 is the solution with the lowest value for function f_1 and the greatest value for f_2 and Sol. 2 is the solution with the greatest value for f_1 and the lowest value

TABLE 1: Sample of solutions.

	f_1	f_2
Sol. 1	252,040	111,920
Sol. 2	345,415	87,246
Sol. 3	255,680	94,599
Sol. 4	290,307	90,737
Sol. 5	308,501	88,061

TABLE 2: Output after the interaction phase.

	f_1	f_2
Sol. 1	284,944	93,621
Sol. 2	285,698	93,423
Sol. 3	286,181	92,835
Sol. 4	287,412	92,244
Sol. 5	288,924	92,024
Sol. 6	289,102	91,641
Sol. 7	289,323	91,063
Sol. 8	289,703	90,954
Sol. 9	290,307	90,737

TABLE 3: Routes of the final solution selected.

	Length (m)	Remaining capacity (kg)
Route 1	75,103.68	64.19
Route 2	92,835.88	141.20
Route 3	92,423.95	99.47
Route 4	25,817.74	2,957.11

for function f_2 . It can be seen that choosing Sol. 2 instead of Sol. 1 leads to a 37.05% deterioration in f_1 and a 22.05% improvement in f_2 . At this point, the procedure required information from the DM about his desired level for each of the two objectives of the problem. To do this, the optimal values of each objective function (ideal points) and a sample of 3 intermediate efficient solutions of the efficient frontier were shown (Sol. 3, Sol. 4, and Sol. 5).

The DM set his desired value for each objective of the reference point; initially $RP = (270000, 90000)$. Using this data, the software obtained a group of 23 efficient solutions. A new sample of five solutions was shown to the DM, and he decided to continue the interaction process by determining a new reference point: $RP = (285000, 91000)$, which leads to a set of 9 efficient solutions. After this iteration, the DM analyzed the last set reached (Table 2), choosing the third solution as the final one. This solution appeared to satisfy the preferences of the manager.

This solution obtained through the interactive process has a total length of 286,181.3 meters, where the longest route implies 92,835.88 meters of it, and it involves four routes. Having a deeper look into the results, the performance of each route is shown in Table 3. With a total distance of 75,103 meters, 10,156 kg loaded from 53 containers visited, Figure 4 displays the tour followed by the first route, whereas the value of the second objective corresponds to the second route (see Figure 5), with a total length of 92,835 meters, 54 containers

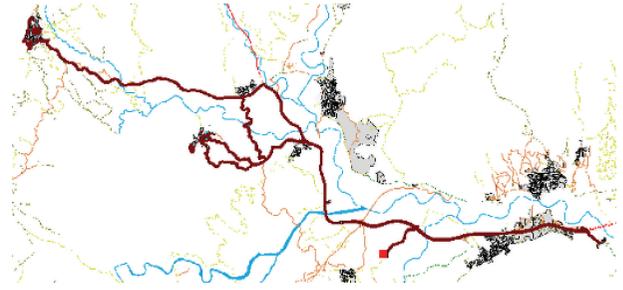


FIGURE 4: Route 1 of the final solution selected by the DM.

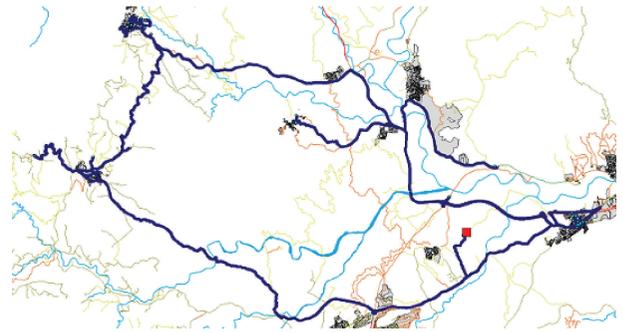


FIGURE 5: Route 2 of the final solution selected by the DM.

visited, and a remaining capacity of 141 kg in the truck. The third route visits 58 containers, loading a 10,121 kg which implies a truck capacity almost full (see Figure 6). Finally, the last and shortest route is displayed in Figure 7. Its length corresponds to the tour that collects 48 containers. Almost 50 km differentiates this route from the second shortest route and its remaining capacity of 2,957 kg.

On one hand, the shortest route coincides with the one ending the process with more capacity available in the truck. On the other hand, the difference between the lengths of the two longest routes is around 400 meters, which is not a very significant distance and the truck is almost full in both cases. However, the third longest route almost achieves the maximum capacity of the truck, so that might have stopped from obtaining a longer route by introducing more containers to collect. These four routes were presented to the DM through the images showed in Figures 4–7, which were generated using the GIS.

4. Conclusions and Future Research

In this paper we address a difficult real-world problem that is the waste collection problem in a city of the southern Spain. This problem was tackled because the company in charge of the waste in the city, was interested in attaining a better planning of this service. One reason for addressing the problem is that, instead of considering a classical objective function, the minimization of the total distance travelled by all vehicles, an additional objective function is also considered. This new objective function is the minimization of the longest route in order to get a balanced set of routes. In this way, a biobjective

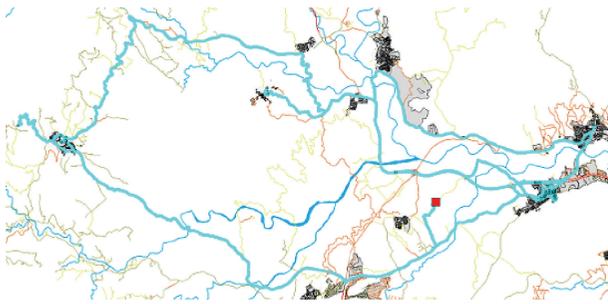


FIGURE 6: Route 3 of the final solution selected by the DM.



FIGURE 7: Route 4 of the final solution selected by the DM.

waste collection problem is addressed. Another important issue is that in most real-world problems more than one cost is associated with each arc; in our particular case, both distance and time costs are considered. Specifically, in the problem of this paper, distance costs are considered in both objective functions and the time costs are included as a constraint of the working time of each vehicle.

The most important feature of this paper is the proposal of a new procedure specifically designed to solve this biobjective real-world problem, modeled as an Asymmetric Capacitated Vehicle Routing Problem with several variations. First, a new algorithm is implemented capable of managing this difficult problem but emphasis is placed on the simplicity, speed, and effectiveness of the algorithm. These characteristics are crucial due to the size of the real-world problem. Additionally, an interactive procedure is also implemented to show different according to DM's preferences.

As future work, we would like to test the proposed framework over larger instances. Also, it could be interesting to improve the propose algorithm to get the set of efficient solution in a single run avoiding the use of single-objective formulations, that is, removing the multiple executions of the ϵ -constraint method. Besides, we would like to improve the interactive phase and to compare it with other methodologies in order to make the whole process easier to the DM.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by research projects (from Universidad de Pablo de Olavide and Universidad de Málaga)

granted by the Spanish Ministry of Science and Innovation. L. Delgado-Antequera and F. Pérez acknowledge support from the Spanish Ministry of Science and Innovation through Project ECO2013-47129-C4-2-R, where Ms. Laura Delgado-Antequera is also part of via BES-2014-068507. A. G. Hernández-Díaz and A. D. López-Sánchez acknowledge support from the Spanish Ministry of Science and Innovation through Project ECO2013-47129-C4-1-R.

References

- [1] E. J. Beltrami and L. D. Bodin, "Networks and vehicle routing for municipal waste collection," *Networks*, vol. 4, no. 1, pp. 65–94, 1974.
- [2] K. Hauge, J. Larsen, R. M. Lusby, and E. Krapper, "A hybrid column generation approach for an industrial waste collection routing problem," *Computers & Industrial Engineering*, vol. 71, no. 1, pp. 10–20, 2014.
- [3] S.-H. Huang and P.-C. Lin, "Vehicle routing-scheduling for municipal waste collection system under the "Keep Trash off the Ground" policy," *Omega*, vol. 55, pp. 24–37, 2015.
- [4] J. R. Gómez, J. Pacheco, and H. Gonzalo-Orden, "A tabu search method for a bi-objective urban waste collection problem," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 1, pp. 36–53, 2015.
- [5] M. Fischetti, P. Toth, and D. Vigo, "A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs," *Operations Research*, vol. 42, no. 5, pp. 846–859, 1994.
- [6] G. Laporte, H. Mercure, and Y. Nobert, "An exact algorithm for the asymmetrical capacitated vehicle routing problem," *Networks*, vol. 16, no. 1, pp. 33–46, 1986.
- [7] R. Martí, V. Campos, M. G. C. Resende, and A. Duarte, "Multiobjective GRASP with path relinking," *European Journal of Operational Research*, vol. 240, no. 1, pp. 54–71, 2015.
- [8] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Computers & Operations Research*, vol. 36, no. 12, pp. 3215–3223, 2009.
- [9] R. Caballero, M. González, F. M. Guerrero, J. Molina, and C. Paralara, "Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1751–1763, 2007.
- [10] L. Tralhão, J. Coutinho-Rodrigues, and L. Alçada-Almeida, "A multiobjective modeling approach to locate multi-compartment containers for urban-sorted waste," *Waste Management*, vol. 30, no. 12, pp. 2418–2429, 2010.
- [11] N.-B. Chang, H. Y. Lu, and Y. L. Wei, "GIS technology for vehicle routing and scheduling in solid waste collection systems," *Journal of Environmental Engineering*, vol. 123, no. 9, pp. 901–910, 1997.
- [12] J. Wang and J. Wright, "Interactive design of service routes," *Journal of Transportation Engineering*, vol. 120, no. 6, pp. 897–913, 1994.
- [13] R. Martí, J. M. Moreno-Vega, and A. Duarte, "Advanced multi-start methods," in *Handbook of Metaheuristics*, vol. 146 of *International Series in Operations Research & Management Science*, pp. 265–281, Springer, 2010.
- [14] Y. Y. Haimes, L. S. Lasdon, and D. Wismer, "Integrated optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 47, pp. 296–297, 1971.

- [15] J. L. Cohon, *Multiobjective Programming and Planning*, Courier Corporation, 2013.
- [16] M. Ehrgott, *Multicriteria Optimization*, vol. 491, Springer, Berlin, Germany, 2013.
- [17] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16, John Wiley & Sons, 2001.
- [18] G. Mavrotas, "Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems," *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009.
- [19] G. Mavrotas and K. Florios, "An improved version of the augmented ϵ -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems," *Applied Mathematics and Computation*, vol. 219, no. 18, pp. 9652–9669, 2013.
- [20] W. Zhang and M. Reimann, "A simple augmented ϵ -constraint method for multi-objective mathematical integer programming problems," *European Journal of Operational Research*, vol. 234, no. 1, pp. 15–24, 2014.
- [21] R. H. Mole and S. R. Jameson, "A sequential route-building algorithm employing a generalised savings criterion," *Operational Research Quarterly*, vol. 27, no. 2, pp. 503–511, 1976.
- [22] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007.
- [23] P. Hansen and N. Mladenović, "An introduction to variable neighbourhood search," in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I. H. Osman, and C. Roucairol, Eds., pp. 433–445, Kluwer Academic Publishers, 1998.
- [24] P. Hansen and N. Mladenovic, "Variable neighborhood search: principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.
- [25] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12, Springer Science & Business Media, Berlin, Germany, 2012.
- [26] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. Coello Coello, and R. Caballero, "g-dominance: reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

