processes

MDPI

Article

Modeling the Municipal Waste Collection Using Genetic Algorithms

Elisabete Alberdi [1,*,†], Leire Urrutia [1,†], Aitor Goti [2,†] and Aitor Oyarbide-Zubillaga [3,†]

1 Department of Applied Mathematics, University of the Basque Country UPV/EHU, 48013 Bilbao, Bizkaia, Spain; leire.urru@gmail.com
2 Deusto Digital Industry Chair, University of Deusto, 48007 Bilbao, Bizkaia, Spain; aitor.goti@deusto.es
3 Department of Industrial Mechanics, Design and Organization, University of Deusto, 48007 Bilbao, Bizkaia, Spain; aitor.oyarbide@deusto.es
* Correspondence: elisabete.alberdi@ehu.eus; Tel.: +34-946-017-790
† These authors contributed equally to this work.

Received: 12 March 2020; Accepted: 21 April 2020; Published: 27 April 2020

check for updates

Abstract: Calculating adequate vehicle routes for collecting municipal waste is still an unsolved issue, even though many solutions for this process can be found in the literature. A gap still exists between academics and practitioners in the field. One of the apparent reasons why this rift exists is that academic tools often are not easy to handle and maintain by actual users. In this work, the problem of municipal waste collection is modeled using a simple but efficient and especially easy to maintain solution. Real data have been used, and it has been solved using a Genetic Algorithm (GA). Computations have been done in two different ways: using a complete random initial population, and including a seed in this initial population. In order to guarantee that the solution is efficient, the performance of the genetic algorithm has been compared with another well-performing algorithm, the Variable Neighborhood Search (VNS). Three problems of different sizes have been solved and, in all cases, a significant improvement has been obtained. A total reduction of 40% of itineraries is attained with the subsequent reduction of emissions and costs.

Keywords: waste collection route planning; traveling salesman problem; genetic algorithms

1. Introduction

The improvement of the waste collection process can be considered aligned with the 11th Sustainable Development Goal (SDG) "Sustainable cities and communities" [1]. Nevertheless, this process is still far from being in an optimal status. As experts state [2], inadequate vehicle routes are among the problems the process should tackle. The optimization of the routes for waste collection vehicles with time window is known as the Waste Collection Vehicle Routing Problem (WCVRP).

As indicated by Caria, Todde, and Pazzona [3], the WCVRP is a specific case of the whole class of problems, known as the Vehicle Routing Problem (VRP). The oldest VRP type problem in the transport history is the Travelling Salesman Problem (TSP), solved for the first time by Lin [4], where the aim is to find the shortest route visiting each member of a collection of locations and returning to the starting point. The TSP has evolved towards solving similar problems with different and additional restrictions and objectives, including the WCVRP presented herein.

The WCVRP needs to organize routes, vehicles, and customers, while respecting the constraints that are imposed by the system. VRP allows for reaching the goals that are referred to as transport logistics, as well as the minimization of costs and carbon dioxide (CO_2) emissions [3]. On top of the overall VRP considerations, WCVRP is concerned with finding cost optimal routes for garbage trucks such that all garbage bins are emptied and the waste is driven to disposal sites while respecting customer time windows and ensuring that drivers are given the breaks that the law requires. The first waste collection problem was probably presented by Beltrami and Bodin [5]. Since then, the problem has evolved into the herein presented WCVRP. Many relevant references have studied this problem so far, offering different approaches for solving the same challenge (or similar ones). Concerning the most relevant and recent ones, Buhrkal, Larsen, and Ropke [6] propose an adaptive large neighborhood search algorithm for solving the problem. Babaee Tirkolaee, Mahdavi, and Seyyed Esfahani [7] apply an improved hybrid simulated annealing algorithm (SA) for optimizing the mathematical model developed. Hannan et al. [8] propose a particle swarm optimization algorithm to optimize a model that considers not only typical distance, total waste, collection efficiency, etc. parameters but also Threshold Waste Level (TWL) tightness factors. De Bruecker et al. [9] present an enhanced model for the WCVRP that models distinct labor cost types and collecting speeds; thus, they differentiate between cheaper regular shifts during traffic peak hours, against higher collection speeds but with expensive, non-regular shift-rates. Rodrigues Pereira Ramos, Soares de Moraisa, and Barbosa-Póvoa [10] study three operational management approaches to define dynamic optimal routes based on sensoring, Radio Frequency Identification (RFID), and considering the access to real-time information on the bins' fill-levels. Benjamin and Beasley [11] develop a model for the waste collection vehicle routing problem with time windows, driver rest period, and multiple disposal facilities as a differential aspect.

It is worth mentioning that the sometimes explicit, but always present idea of this optimization problem is the reduction of CO_2 emissions, as the current logistic methods used in this routing problems depend strongly on fossils fuels [12].

The fact of not having generalized optimized routes for the WCVRP is striking, as many solutions for this process can be found in the literature. When analyzing the potential causes of this breach between the practitioners and the academia, it could be stated that generic routing tools are considered incomplete for this variant of the traveling salesman problem [13], whereas more specific tools (such as [14,15]) have a related cost that often entities are not willing to afford. Additionally, many researchers developed their solutions to the problem, but normally they are more focused on publishing them than in offering them to entities that manage the waste collection. Thus, even though the problem has been deeply studied from a theoretical point of view, its application to real-world problems is scarce.

Within this context, the aim of this work has been to show the development and implementation of a procedure for the optimization for a local commonwealth for tackling the WCVRP of a region.

In the final solutions, Genetic Algorithms (GAs) have been applied. GAs belong to the more general classification of evolutionary optimization techniques or evolutionary programs and they are surely the most widely known type of Evolutionary Algorithms. They are based on selection, crossover, and mutation principles of Darwin's theory of evolution. In the last few years, there has been a growing effort to apply GAs to general constrained optimization problems as most of engineering optimization problems often see their solution constrained by a number of restrictions imposed on the decision variables [16].

The research method had three major steps: first, we performed a literature review of optimization algorithms applied to this case that is detailed in Section 1. The aim of this study was to analyze how this problem was modeled in the past, and to pre-select which types of algorithms could fit best to the end user requirements in terms of user friendliness, cost, and quality of results. Second, the characteristics of this specific optimization case (detailed in Section 2) were analyzed to offer the company a selection of the algorithm fitting best to its requirements. Finally, the designed solution was successfully implemented and tested with the pre-selected types of algorithms, to choose the one performing best, as it is detailed in Section 3 Methodology, Section 4 Algorithm, and Section 5 Results, remarking that the obtained results raised the interest of the surrounding communities.

2. Optimization Problems

Combinatorial optimization problems can be written mathematically as:

$$\text{minimize } f(x), \tag{1}$$
$$\text{subject to} \quad h_i(x) \geq a_i, \quad i = 1, \ldots, m, \tag{2}$$
$$f_j(x) = b_j, \quad j = 1, \ldots, s, \tag{3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $h_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ and $f_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, s$ are the constraints.

The optimization problem has been written as a minimization problem, but after some modifications it can be written as a maximization problem:

$$\min g(x) = \max -g(x). \tag{4}$$

In the same way, the equality constraints $f_j(x) = b_j$ can be written as inequalities:

$$f_j(x) \geq b_j \text{ and } f_j(x) \leq b_j. \tag{5}$$

The simplest constrained optimization problem arises when the objective function $f(x)$ and the restrictions are linear functions. This type of problem is a linear programming problem, and it can be solved quite efficiently by the simplex algorithm. However, in the majority of the optimization problems, neither the objective function nor the restrictions are linear functions. The vast majority of these problems are NP-complete problems, which means that there is no any solving algorithm that can be executed in polynomial time in relation to the size of the problem. In complexity theory, NP-complete denotes the set of problems that are not solvable by a deterministic polynomial time algorithm. The feasible solutions' space is so large that the computation of the exact solution requires a lot of time. NP-complete problems can be solved by a restricted class of brute force search algorithms and they can be used to simulate any other problem with a similar algorithm. Genetic algorithms are also a good and efficient choice to find an approximate solution of these problems [17]. A classical NP-complete problem is the Travelling Salesman Problem (TSP), in which the shortest route for a traveling salesman starting and finishing in the same point and visiting every city once has to be found.

An efficient way to solve these types of problems is using genetic algorithms. The basic principles of genetic algorithms were established by Holland [18], and they are well described in several texts [19–23]. Owing to their simplicity, flexibility, ease of implementation, minimal requirements, fast convergence towards close-to-optimal solutions, and global perspective, GAs are successfully used in a wide variety of problems [16]. As these characteristics are essential for practitioners to have a utilizable solution, and as the performed literature review did not bring better solutions, the chosen option was to implement the simple GA (SGA) presented herein.

Traveling Salesman Problem

In the TSP problem, a collection of n cities are given. The objective is to determine the shortest route that a salesman has to follow, in which each city is visited once and then the salesman returns to the starting point of the route. This problem can be defined mathematically as follows:

Given an integer n and an $n \times n$ matrix $D = (d_{ij})$ in which each d_{ij} is the distance between two cities, the cyclic permutation π of the integers $i = 1, 2, \ldots, n$ that minimizes the distances has to be determined. As a first approximation, the feasible search space is formed by all cyclic permutations of the numbers $\{1, 2, \ldots, n\}$:

$$F = \left\{ (\pi_1 \pi_2 \ldots \pi_n) \middle| \, \pi_i \in \{1, 2, \ldots, n\} \; and \; \pi_i \neq \pi_j \forall i \neq j \right\}. \tag{6}$$

The number of elements of this space is $n!$, being n the number of cities. In this way, the length of a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ can be expressed as:

$$\sum_{i=2}^{n} d_{\pi_{i-1},\pi_i} + d_{\pi_n,\pi_1}. \tag{7}$$

For each permutation π, there are $(n-1)$ permutations that starting in a different city are similar to the given one. That is to say, the distance of these n permutations is the same. Taking into account this consideration, the size of the feasible space is $n!/n = (n-1)!$. Moreover, if the distance matrix D is symmetric, the distance of each permutation in both directions is the same, and the size of the feasible space is reduced to $(n-1)!/2$. As it can be observed, the main difficulty of this problem is the huge number of possible tours.

3. Methodology

The objective of this work is to find an optimal itinerary for the waste collection. This methodology will be applied to the data of Sopelana, a municipality in the province of Biscay, autonomous community of the Basque Country (Spain). Sopelana is located in the region of Plencia-Munguia or Uribe, and it is part of the Commonwealth of Services of Uribe Coast. It has an extension of 8.40 km^2 and a population of 13,510 inhabitants, a figure that in summer is usually multiplied by four due to summer visitors.

It is the responsibility of the commonwealth everything related to waste management. There are various trash cans to collect waste:

- Restwaste: 317 trash cans. They are distributed in three routes. There are 186 trash cans in the first route, 72 trash cans in the second route, and 59 in the third one. The trash of the first route is collected everyday. The second route is done four days per week and the third route three days per week.
- Organic waste: 29 trash cans. This waste is collected once per week.
- Small recipients of plastic and metal: 85 trash cans. This waste is collected twice per week.
- Glass: 69 trash cans. Glass is collected once per month.
- Paper: 62 trash cans. These trash cans are divided in two groups depending on their filling frequency. Paper is collected everyday, but not all the containers are emptied everyday.
- Oil: 4 trash cans. This waste is collected twice per month.
- Reusable waste: 7 trash cans. From September to June, it is collected once per week, and in July and August twice per week.
- Batteries: 31 trash cans. They are collected twice per month.

Other types of waste such as big volume wastes are collected once per week following the same route as for the restwaste; there are eight points to collect lamps and fluorescent lightings and they are collected when the container is full; there are some locations in which CDs are collected when the containers are full. All the aforementioned data were given by the local government.

In this work, three problems of different sizes will be analyzed. In the three cases, the aim is to improve the waste collection itinerary in the sense of obtaining a shorter route than the one followed nowadays. A small problem of reusable waste with seven trash cans, a medium problem of organic waste consisting of 29 trash cans, and a big problem consisting of the first route of restwaste with a total of 186 trash cans will be considered. In the case of restwaste, there are several trash cans in the same location. Specifically, the 186 trash cans are distributed in 147 different locations; thus, these locations will be considered in the problem. The data of these problems can be seen in Table 1.

Table 1. Three problems analyzed.

Waste-Type	Number of Locations	Collecting Frequency
Reusable waste	7	September to June once per week, July and August twice per week
Organic waste	29	Once per week
Restwaste	147	Everyday

According to the data given by the enterprise in charge of restwaste collection, between 8000 kg and 13,000 kg of restwaste are collected everyday in Sopelana. A truck is enough to carry out this collection. The truck used has a load of 13 tons, and it has a compaction mechanism. Each container has 1.1 m^3, which is reduced to 0.183 m^3 after compaction. This means that the volume of 125 trash cans that are full can fit in the truck (125×0.183 m^3 = 22.875 m^3). In the event that the truck was filled during the collection of the 186 containers (147 location points), it would be emptied in the dump intended for it and, after that, it would continue with the route. However, this event is not very probable as it means that 125 trash cans out of 186 are full up (which is the 67.20%).

4. Algorithms

For the three problems, the coordinates of the locations and the distance matrices have been calculated using Google Maps (https://www.google.com/maps, data from October and November 2019). The coordinates of all locations and the distance matrices can be found in Appendix A. The smallest distance between two locations has been considered in the distance matrices. The distance matrices are not symmetric, as the way back and forth from a location to another may be different.

The three problems have been solved using a VNS and a GA. The smallest problem (reusable waste, seven locations) has also been solved using the brute force algorithm. These two algorithms were chosen among all the set of alternatives for developing the final solution for maintainability reasons. The developers of the solutions are academic institutions, so we cannot provide maintenance services. As both literature [24] and the news [25] state, there is a lack of availability of computer science technicians. Because of this fact, the final user of the solution wanted a competitive but mainly easy to maintain algorithm. These algorithms being two of the most frequently applied when developing the syllabuses of subjects related to programming, the research team decided to compare them and offer the most effective possible solution. The distances that correspond to all the permutations of $n = 7$ elements have been calculated that is $7! = 5040$. The process has been implemented as it is explained in Algorithm 1.

Algorithm 1 Algorithm to determine solutions using brute force.

```
procedure BRUTEFORCE( distance matrix of n = 7 problem )
    Create all the permutations of n = 7 elements
    for i=1:7! do
        Calculate the distance of the permutation
    end for
    Calculate the shortest distance among all the permutations
end procedure
```

We have started by trying the VNS algorithm [26] for all the problems. The VNS consists of applying a local search method repeatedly in the neighborhood $\mathcal{N}_k$ of the actual solution. When a local optimal is reached, the algorithm changes the system of neighborhood with the aim of escaping from local optima and reaching a better one. For this reason, it can be said that the VNS performs well both when searching local and global optima. In Algorithm 2, the process followed is explained. We have implemented two systems of neighborhoods: the 2-opt neighborhood and a swap-based neighborhood. The neighborhood structure 2-opt consists of changing a pair of edges between cities [27]. The swap-based neighborhood that we have created swaps the first element of the permutation with all the rest.

Algorithm 2 Algorithm to determine solutions using VNS.

```
procedure VARIABLE NEIGHBORHOOD SEARCH(distance matrix)
    Choose a set of neighborhood structures N_k for k = 1, 2, ..., k_max
    Generate the initial solution
    Consider the initial solution as the best one
    k ← 1
    while k ≤ k_max do
        while There is an improvement do
            Choose the neighborhood system that corresponds to k
            Find the best solution among all the neighbors
            if The solution is improved then
                Update the best solution and its evaluation function value
                Continue to search with k ← 1
            else
                Continue to search with k ← k + 1
            end if
        end while
    end while
    Output the shortest distance and the corresponding route of the overall process
end procedure
```

Additionally, for all the problems $n = 7$, $n = 129$, and $n = 147$, another two algorithms have been implemented. In Algorithm 3, an initial population of m different individuals (permutations) has been created. The evaluation function (distance of the route) of each individual has been calculated. In each generation, the process of selecting two parents randomly, the crossover operator, the posterior correction of the individual in order to be a permutation, and the mutation process have been performed $m/2$ times. If the new descendants are different from the individuals of the population and if their evaluation function (fitness function) is smaller than the worst (the largest) of the population, the worst individuals are replaced by these new descendants. The fourth algorithm implemented only differs from the third in the fact that the routes followed nowadays in the trash collection (one route in each problem) are inserted as a seed in the initial population.

Algorithm 3 Algorithm to determine solutions using SGA.

```
procedure SIMPLE GENETIC ALGORITHM(distance matrix, generations, population size, probabilities)
    Create an initial population of m different permutations randomly
    Compute the evaluation function of each permutation
    for i=1:generations do
        for j=1:m/2 do
            Select two parents randomly from the population
            Cross with a certain probability to produce two descendants
            Correct the descendants to be permutations
            Mutate each individual with a certain probability
            Compute the evaluation function of each descendant
            if evaluation function of the descendants smaller than the largest evaluation then
                if the descendants are not repeated in the population then
                    Replace the descendants by the permutations with largest evaluation
                end if
            end if
        end for
        Output the shortest distance and the corresponding route of each generation
    end for
    Output the shortest distance and the corresponding route of the overall process
end procedure
```

The crossover operator used is the classical one [18]. A crossover point is selected randomly from which the two strings of the parents are broken into separate parts. The new descendants are formed by recombination of these parts. For example, consider $n = 7$ and the routes:

$$(1\ 2\ 3\ 4\ 5\ 6\ 7)$$
$$(2\ 4\ 5\ 3\ 7\ 1\ 6)$$

Randomly, a crossover point in which the strings are broken into separate parts is selected. Suppose that the crossover point is chosen between the third and the fourth bit:

$$(1\ 2\ 3\ \mid 4\ 5\ 6\ 7)$$
$$(2\ 4\ 5\ \mid 3\ 7\ 1\ 6)$$

Combining the head of the first route with the tail of the second route and vice versa, the result is the following:

$$(1\ 2\ 3\ \mid 3\ 7\ 1\ 6)$$
$$(2\ 4\ 5\ \mid 4\ 5\ 6\ 7)$$

If the resulting descendants are not permutations, then a correction algorithm is applied in order to obtain two individuals that belong to the feasible space. Each resulting individual is repaired, by calculating the repeated values and their positions, and by inserting the missing values randomly on the positions where the repeated values are located.

The exchange mutation operator, also called the swap mutation operator, has been applied, in which two positions of the route are selected randomly and the cities on those positions are exchanged [28]—for example, if we consider the route:

$$(1\ 2\ 3\ \mathbf{5}\ 7\ 4\ \mathbf{6}),$$

and if we choose the fourth and seventh positions, the cities on those positions are interchanged:

$$(1\ 2\ 3\ \mathbf{6}\ 7\ 4\ \mathbf{5})$$

5. Results

In this section, the results obtained for each of the problems are presented.

For $n = 7$, the brute force algorithm has been applied first. The shortest itinerary has 7.67 km and the longest 13.2 km. In Figure 1, the distances of all permutations ($7! = 5040$) are shown. All the permutations are represented in the horizontal axis (that is, $7! = 5040$), and the distance that corresponds to each of them is plotted in the vertical axis.

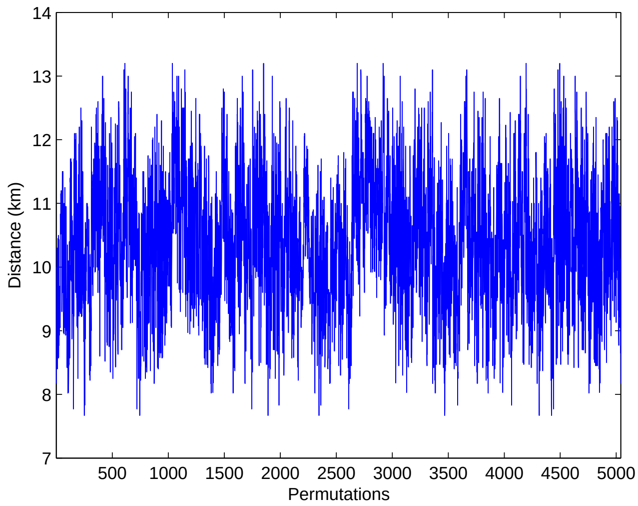

Figure 1. Brute force application, problem $n = 7$.

For the three problems, the VNS algorithm has been applied next. Ten executions have been done for each problem. The best, the worst, the mean, and the median distances of this executions are presented in Tables 2–4. In all the executions, an initial solution has been generated randomly.

Table 2. Results of 10 executions $n = 7$ problem, VNS.

$n = 7$	VNS
Best distance (km)	7.6700
Worst distance (km)	8.4000
Mean distance (km)	7.7690
Median distance (km)	7.6700

Table 3. Results of 10 executions $n = 29$ problem, VNS.

$n = 29$	VNS
Best distance (km)	19.4170
Worst distance (km)	25.5570
Mean distance (km)	22.5475
Median distance (km)	21.9450

Table 4. Results of 10 executions $n = 147$ problem, VNS.

$n = 147$	VNS
Best distance (km)	61.5470
Worst distance (km)	71.3560
Mean distance (km)	66.8944
Median distance (km)	66.9507

In addition, finally, the SGA (without and with a seed) has been performed. For $n = 7$, taking into account that the average of the shortest and the longest route is 10.4350 km, the route $(6, 7, 3, 4, 2, 5, 1)$ of 10.45 km has been chosen as seed. In the cases of $n = 29$ and $n = 147$, the actual itineraries have been chosen as seed. These data are available in Table 5. Notice that, if the VNS algorithm is started, taking the actual itinerary as initial solution, all the executions have the same performance. In this case, for $n = 7$, a route of 8.2500 km is obtained; for $n = 29$, a route of 18.7670 km; and, for $n = 147$, a route of 34.4280 km.

Table 5. Average distance route ($n = 7$) and actual routes ($n = 29$, $n = 147$).

	$n = 7$	$n = 29$	$n = 147$
Route	(6, 7, 3, 4, 2, 5, 1)	(26, 24, 23, 18, 15, 11, 12, 13, 16, – 17, 20, 21, 22, 19, 9, 7, 27, 28, – 14, 25, 10, 8, 6, 2, 3, 4, 5, 29, 1)	(135, 136, 140, 141, 139, 138, 137, 107, 142, 147, – 110, 42, 40, 44, 101, 102, 104, 146, 105, 103, – 108, 109, 144, 143, 106, 145, 111, 112, 115, 113, – 41, 114, 116, 121, 126, 127, 132, 131, 133, 134, – 130, 129, 128, 125, 120, 118, 119, 117, 124, 122, – 123, 58, 57, 69, 70, 96, 97, 100, 99, 98, – 68, 67, 66, 49, 48, 47, 45, 46, 55, 56, – 65, 64, 62, 95, 94, 77, 76, 75, 74, 73, – 61, 60, 59, 36, 35, 37, 84, 85, 88, 89, – 86, 7, 10, 9, 8, 3, 2, 1, 4, 6, – 5, 11, 14, 13, 15, 16, 18, 17, 12, 90, – 91, 92, 93, 78, 79, 80, 81, 87, 19, 83, – 82, 71, 29, 27, 26, 28, 23, 22, 21, 20, – 24, 25, 72, 30, 31, 32, 34, 33, 38, 43, – 50, 52, 63, 54, 53, 51, 39)
Distance (km)	10.4500	22.9170	46.2890

An execution for each problem has been performed. Parameters have been chosen according to [29]: a relatively high crossover rate (≥ 0.6), small mutation rate (range $[0.001, 0.1]$) and a moderate population size. Data and results are presented in Tables 6–8. In all the cases, the execution in which the seed is considered obtains better solutions.

Table 6. Results of the $n = 7$ problem.

$n = 7$	SGA without Seed	SGA with Seed
Generations	20	20
Population size	30	30
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best route	(3, 2, 1, 6, 7, 4, 5)	(7, 4, 5, 3, 2, 1, 6)
Best distance (km)	7.6700	7.6700

Table 7. Results of the $n = 29$ problem.

$n = 29$	SGA without Seed	SGA with Seed
Generations	4000	4000
Population size	200	200
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best route	(18, 16, 28, 27, 14, 25, 13, 12, 11, 9, – 7, 4, 3, 2, 5, 8, 6, 10, 17, 20, – 21, 23, 24, 26, 22, 19, 1, 29, 15)	(10, 12, 13, 17, 23, 24, 26, 22, 19, 21, – 20, 18, 15, 11, 9, 7, 28, 27, 14, 25, – (16, 1, 29, 5, 4, 3, 2, 8, 6)
Best distance (km)	18.0270	17.6570

Table 8. Results of the $n = 147$ problem.

$n = 147$	SGA without Seed	SGA with Seed
Generations	4000	4000
Population size	400	400
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best route	(102, 103, 142, 113, 114, 125, 130, 134, 133, 132, − 124, 117, 75, 61, 79, 82, 87, 21, 18, 19, − 53, 139, 141, 52, 36, 85, 8, 7, 10, 9, − 88, 76, 48, 46, 57, 50, 49, 47, 45, 121, − 120, 119, 116, 56, 64, 62, 99, 98, 69, 67, − 66, 122, 123, 129, 131, 128, 126, 55, 39, 71, − 73, 97, 68, 65, 63, 104, 146, 105, 137, 138, − 54, 20, 28, 32, 34, 41, 42, 33, 38, 31, − 29, 24, 25, 30, 35, 44, 106, 107, 143, 147, − 111, 112, 115, 144, 108, 127, 118, 95, 77, 83, − 3, 4, 2, 1, 14, 80, 110, 43, 51, 100, − 70, 74, 59, 58, 27, 26, 12, 17, 6, 5, − 22, 23, 72, 78, 86, 11, 13, 15, 16, 81, − 37, 96, 94, 84, 89, 90, 91, 93, 92, 60, − 40, 101, 145, 140, 135, 136, 109)	(141, 140, 135, 136, 107, 138, 137, 139, 142, 147, − 41, 42, 40, 101, 44, 102, 104, 146, 105, 108, − 145, 109, 144, 143, 106, 103, 111, 112, 110, 113, − 115, 114, 116, 51, 126, 121, 125, 129, 130, 134, − 133, 131, 128, 127, 119, 132, 124, 117, 120, 122, − 123, 50, 118, 69, 64, 62, 68, 100, 99, 98, − 97, 67, 66, 49, 48, 47, 45, 46, 55, 43, − 56, 70, 96, 95, 94, 77, 78, 73, 75, 74, − 61, 60, 59, 39, 31, 37, 76, 84, 88, 86, − 90, 10, 9, 8, 7, 3, 2, 4, 1, 6, − 5, 12, 11, 13, 15, 17, 18, 16, 14, 89, − 91, 92, 93, 85, 87, 80, 79, 81, 19, 82, − 83, 71, 29, 27, 25, 26, 23, 22, 21, 20, − 24, 28, 72, 30, 32, 35, 34, 33, 38, 57, − 58, 65, 63, 54, 53, 52, 36)
Best distance (km)	48.3510	30.4150

The longest, the shortest, and the average-distance itineraries for $n = 7$ can be seen in Figure 2. The limit of the municipality of Sopelana is marked using a thick red line.

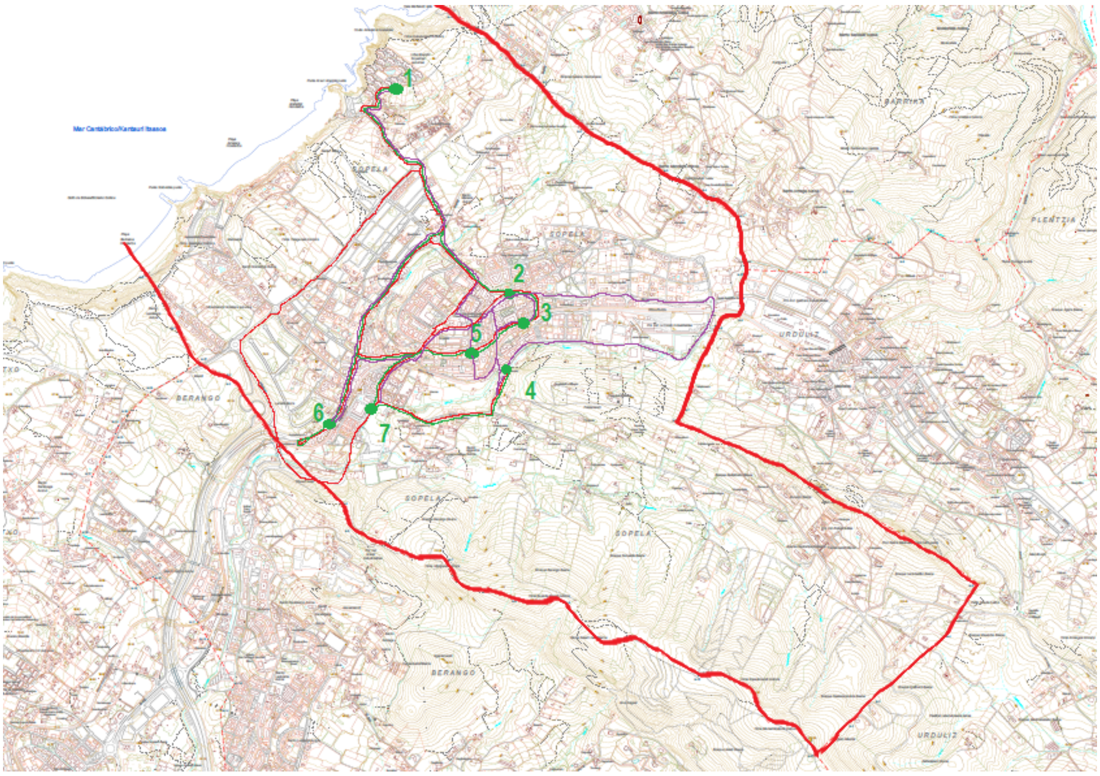

Figure 2. Routes of problem $n = 7$, longest in red, shortest in green, "average" in purple.

Additionally, several executions have been performed for each of the problems ($n = 7, 29, 147$). The best, the worst, the mean, and the median distance of this performance are presented in Tables 9–11. The values of these tables are acquired after performing 10 executions for each version of each problem.

Table 9. Results of 10 executions $n = 7$ problem, SGA.

$n = 7$	SGA without Seed	SGA with Seed
Generations	20	20
Population size	30	30
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best distance (km)	7.6700	7.6700
Worst distance (km)	8.0200	7.7700
Mean distance (km)	7.7050	7.6900
Median distance (km)	7.6700	7.6700

Table 10. Results of 10 executions $n = 29$ problem, SGA.

$n = 29$	SGA without Seed	SGA with Seed
Generations	4000	4000
Population size	200	200
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best distance (km)	17.3770	17.1870
Worst distance (km)	19.0170	18.3670
Mean distance (km)	18.1910	17.8530
Median distance (km)	17.9420	17.7420

Table 11. Results of 10 executions $n = 147$ problem, SGA.

$n = 147$	SGA without Seed	SGA with Seed
Generations	4000	4000
Population size	400	400
Crossover probability	0.8	0.8
Mutation probability	0.01	0.01
Best distance (km)	48.4740	29.6310
Worst distance (km)	54.6080	32.7670
Mean distance (km)	50.9096	31.7165
Median distance (km)	50.3485	31.9552

For the largest problems, $n = 29$ and $n = 147$, computation has been repeated choosing different parameters in order to obtain better results. The most important change is that a larger population size has been used, i.e., 4000 individuals. Execution data and results can be seen in Tables 12 and 13.

Table 12. Results of the $n = 29$ problem.

$n = 29$	SGA without Seed	SGA with Seed
Generations	800	800
Population size	4000	4000
Crossover probability	0.8	0.8
Mutation probability	0.8	0.8
Best route	$(16, 17, 23, 24, 26, 22, 19, 21, 20, 18, -$ $15, 11, 10, 12, 2, 1, 29, 5, 4, 3-$ $8, 6, 9, 7, 28, 27, 14)$	$(15, 11, 10, 13, 16, 17, 23, 24, 26, 22, -$ $19, 21, 20, 18, 14, 25, 28, 27, 12, 9, -$ $(-, 7, 4, 8, 6, 3, 5, 2, 1, 29)$
Best distance (km)	17.1700	16.9370

Table 13. Results of the $n = 147$ problem.

$n = 147$	SGA without Seed	SGA with Seed
Generations	1800	1800
Population size	4000	4000
Crossover probability	0.8	0.8
Mutation probability	0.8	0.8
Best route	(107, 142, 53, 37, 73, 60, 59, 50, 58, 67, − 66, 29, 26, 25, 24, 28, 30, 51, 121, 116, − 114, 115, 56, 36, 35, 33, 34, 46, 122, 124, − 117, 120, 126, 127, 132, 119, 19, 80, 17, 16, − 77, 78, 71, 76, 94, 82, 88, 85, 84, 39, − 31, 32, 38, 57, 143, 105, 146, 104, 103, 108, − 147, 136, 135, 112, 111, 79, 87, 81, 22, 23, − 72, 48, 45, 145, 144, 139, 138, 137, 123, 128, − 134, 133, 110, 43, 113, 41, 40, 42, 141, 140, − 64, 62, 68, 69, 70, 96, 95, 93, 92, 91, − 10, 3, 1, 2, 4, 83, 97, 65, 63, 61, − 75, 74, 118, 100, 99, 98, 49, 47, 55, 125, − 129, 130, 131, 54, 27, 20, 21, 18, 12, 11, − 6, 5, 7, 8, 9, 89, 90, 86, 14, 13, − 15, 52, 109, 44, 101, 102, 106)	(141, 140, 135, 136, 107, 138, 137, 139, 142, 112, − 41, 42, 40, 101, 44, 102, 104, 146, 105, 106, − 103, 108, 144, 143, 145, 109, 147, 110, 43, 113, − 115, 114, 116, 123, 127, 126, 129, 130, 134, 133, − 132, 125, 128, 131, 121, 122, 124, 117, 120, 119, − 52, 118, 100, 69, 70, 94, 73, 97, 99, 98, − 68, 67, 66, 49, 48, 47, 45, 46, 55, 56, − 58, 64, 62, 96, 95, 77, 78, 71, 61, 75, − 74, 60, 59, 36, 35, 37, 76, 84, 88, 86, − 90, 10, 9, 8, 7, 3, 1, 2, 4, 6, − 5, 11, 12, 13, 15, 17, 16, 18, 14, 89, − 91, 92, 93, 85, 79, 87, 80, 81, 19, 83, − 82, 30, 29, 27, 25, 26, 23, 22, 21, 20, − 24, 28, 72, 39, 31, 32, 34, 33, 38, 57, − 50, 65, 63, 54, 53, 51, 111)
Best distance (km)	40.3460	27.0950

The actual itineraries are presented in Figures 3 and 4, and the improved itineraries of Tables 12 and 13 obtained with the new parameters in Figures 5 and 6. In the case of problem $n = 147$, Figures 4 and 6, the first and the last location of the route are marked in red.

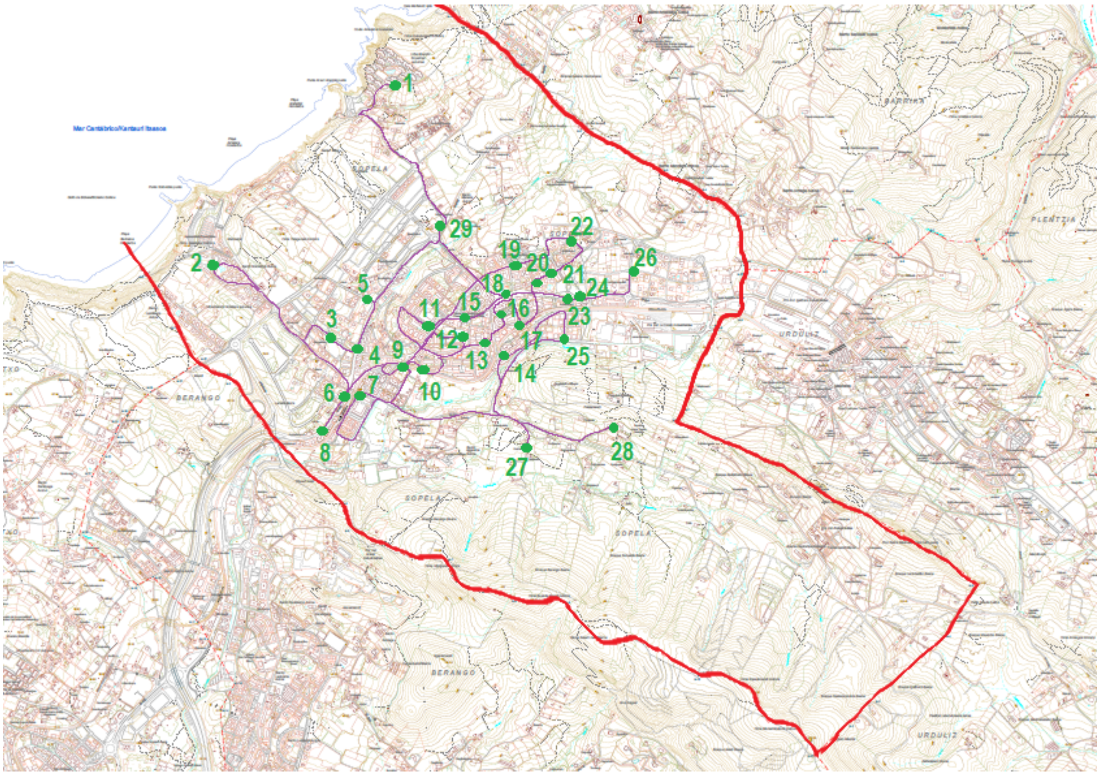

Figure 3. Actual route of problem $n = 29$.

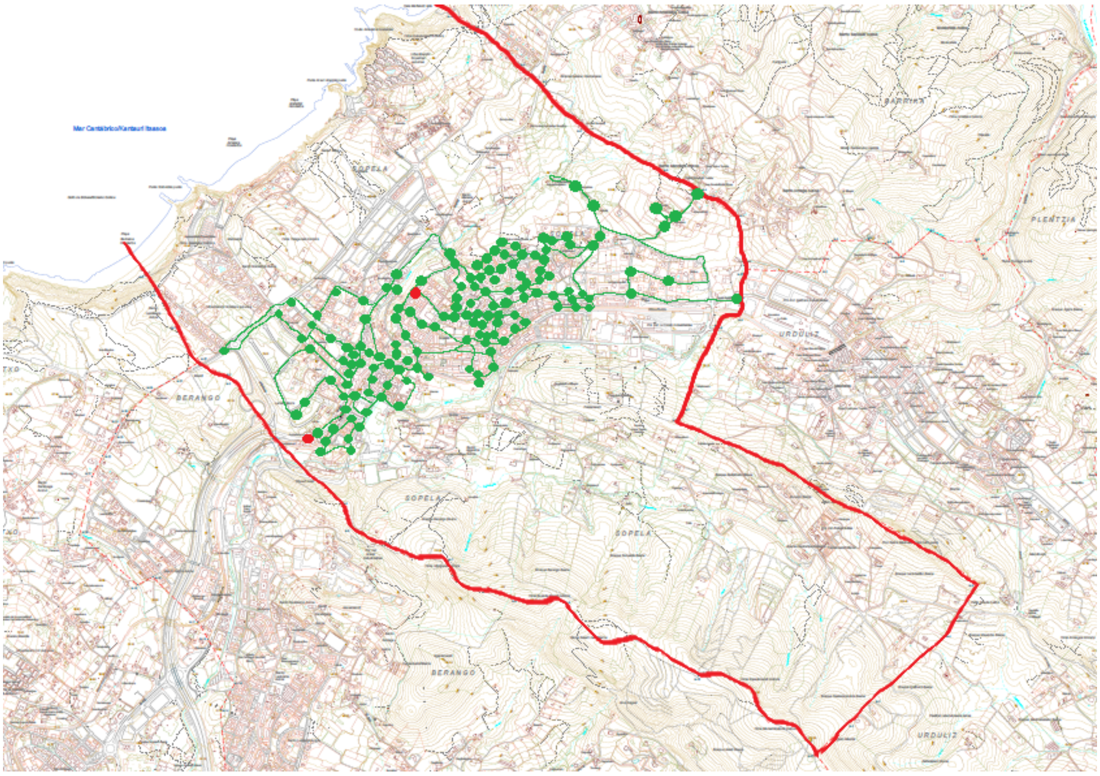

Figure 4. Actual route of problem $n = 147$.

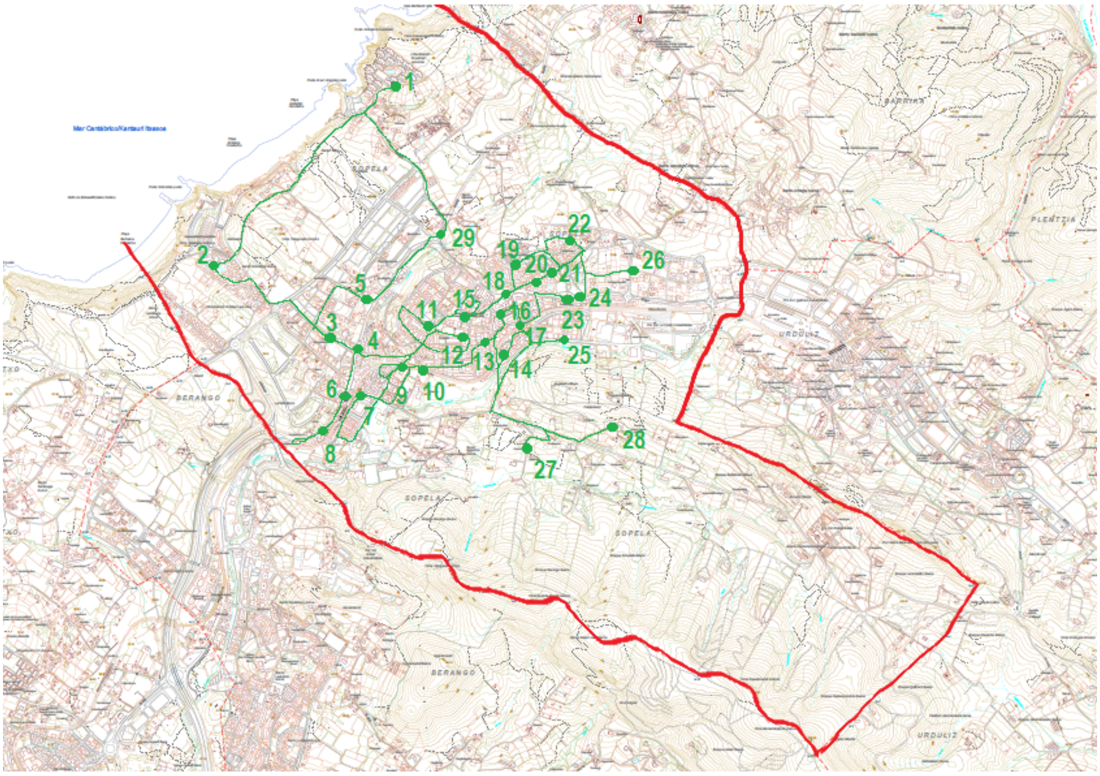

Figure 5. Smallest route obtained for problem $n = 29$.

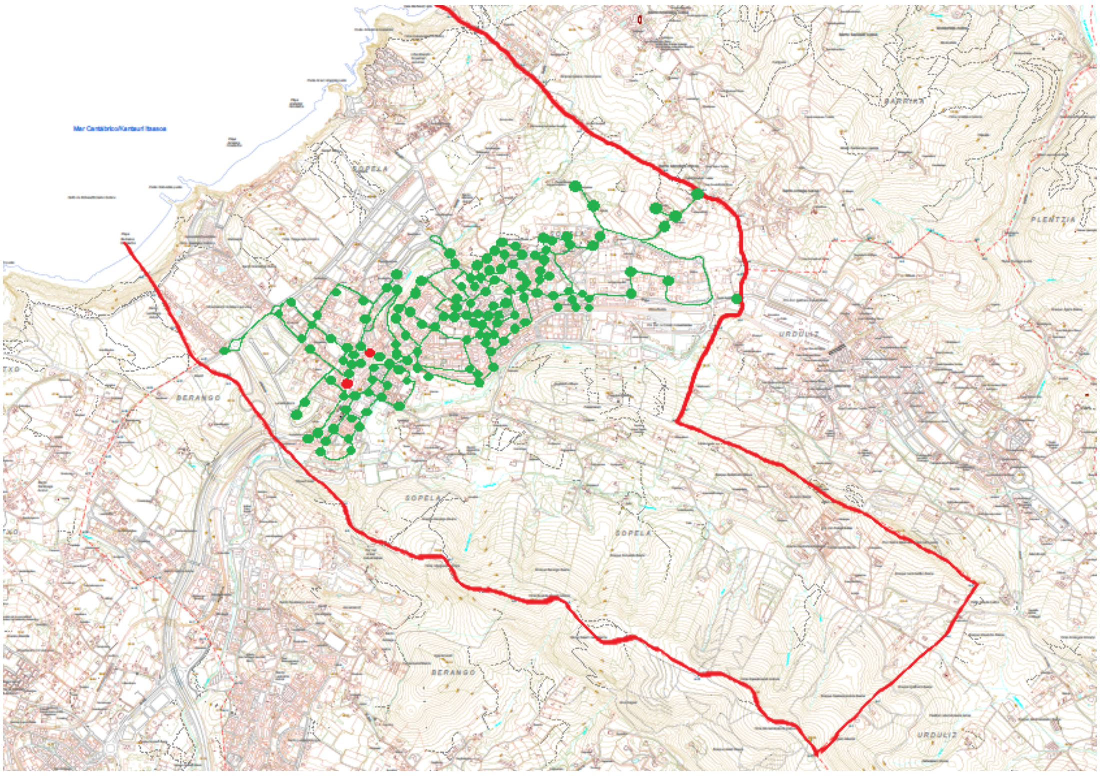

Figure 6. Smallest route obtained for problem $n = 147$.

6. Conclusions

In this work, three waste collection itineraries have been improved in a municipality of Biscay (Spain). The average itinerary of the reusable waste ($n = 7$ problem) has had a reduction of 2.78 km; and the actual itineraries of organic waste ($n = 29$ problem) and restwaste ($n = 147$ problem) have been reduced 5.98 km and 19.194 km, respectively. Taking into account the collection frequencies of these three itineraries, this makes a total reduction of 7400 km per year, that is to say, a reduction of the 40% of the total actual itineraries.

The truck has a continent of 13 tons and it has a compaction mechanism. Considering the following average data for the truck: vehicle of 26 Tn, speed limit 30 km/h, with 270 CV minimum engine power (1 CV = 735.39, 875 W= 0.986 HP) and diesel fuel type and 29 L/100 km consumption [30]. This implies a reduction of 5.58 Tn of CO_2 emissions, 0.43 Kg of CO, and 13.95 Kg of NOx per year. In addition to these improvements, a direct cost savings of 7294€ was obtained (considering the direct cost per kilometer calculation model recommended by the Ministry of Transportation, Logistics and Urban Agenda of Spain [31]).

It is worth mentioning that the research team performed an additional validation for the developed model. Specifically, the actual consumptions versus the ones proposed by the model were analyzed at the two comparable routes (the ones the truck made before and after the optimization), obtaining negligible differences. This double check on the results and the easiness of the solution have raised the interest of other commonwealths, such as the one of Lea-Artibai. Thus, the short and midterm future steps would be oriented to the application of the same procedure to other local communities, incorporating other parameters such as the elevation information of the routes.

Author Contributions: E.A., writing, conceptualization, and investigation; L.U., conceptualization and investigation; A.G., investigation, writing and funding acquisition and A.O.-Z., writing and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundación BBK, partner of the Deusto Digital Industry Chair.

Acknowledgments: We would like to thank the partners of the Deusto Digital Industry Chair (Etxe-Tar, General Electric, Idom, Accenture, Fundación Telefónica, Fundación BBK) the interest and support shown during this research. We would like also to thank the municipality of Sopelana and the commonwealth of the regions of Lea-Artibai for placing their trust and confidence in our organization.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GA	Genetic Algorithm
VNS	Variable Neighborhood Search
SDG	Sustainable Development Goal
WCVRP	Waste Collection Vehicle Routing Problem
VRP	Vehicle Routing Problem
TSP	Travelling Salesman Problem
SA	Simulated Annealing
TWL	Threshold Waste Level
RFID	Radio Frequency Identification
NP	Nondeterministic Polynomial Time
SGA	Simple Genetic Algorithm

Appendix A. Data

Table A1. Coordinates of the seven locations.

Location	Latitude (Decimal Degrees)	Longitude (Decimal Degrees)
1	43.391464	−2.987950
2	43.381466	−2.980558
3	43.380186	−2.979504
4	43.377695	−2.980651
5	43.378805	−2.982968
6	43.375206	−2.992354
7	43.374066	−2.990935

Table A2. Distances of the seven locations in kilometers.

From ↓ To →	1	2	3	4	5	6	7
1	0 1.7	1.9	2.2	2.2	2.3	2.4	
2	1.7	0	0.23	0.55	0.55	1.7	1.1
3	1.9	0.27	0	0.8	0.8	2	1.3
4	3.4	1.8	0.5	0	0.4	1.9	0.95
5	2	0.4	0.4	0.4	0	1.2	0.8
6	2.4	1.7	1.9	1.9	1.2	0	1
7	3.8	1.7	2	1.6	1.6	1.7	0

Table A3. Coordinates of the 29 locations.

Location	Latitude (Decimal Degrees)	Longitude (Decimal Degrees)
1	43.391326	−2.988336
2	43.382808	−2.999907
3	43.379497	−2.992421
4	43.378769	−2.990454
5	43.381233	−2.989976
6	43.376588	−2.991162
7	43.376443	−2.990605
8	43.374686	−2.993086
9	43.377947	−2.987348
10	43.377956	−2.986103
11	43.379873	−2.985665
12	43.379364	−2.983446
13	43.378979	−2.982283
14	43.378942	−2.980098
15	43.380483	−2.982809
16	43.380490	−2.980981
17	43.380161	−2.979534
18	43.381520	−2.980494
19	43.382913	−2.979929
20	43.382040	−2.978645
21	43.382416	−2.977694
22	43.384166	−2.977144
23	43.381216	−2.976394
24	43.381311	−2.975217
25	43.379194	−2.977072
26	43.382615	−2.972005
27	43.374091	−2.979216
28	43.375154	−2.973282
29	43.384723	−2.984886

Table A4. Distances of the 29 locations in kilometers.

From ↓ To →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	2.2	1.9	1.9	1.7	3	2.5	2.4	2.4	2.1	1.9	2.3	2.2	2.1	1.8	2	1.9	1.7	1.7	1.9	2	2.4	2	2.2	2.5	2.5	2.6	3	1.1
2	1.5	0	0.9	1.2	1.1	2	1.9	1.4	1.8	1.6	1.7	1.9	1.9	2.8	2.4	2.3	2.3	2.2	2.3	2.3	2.4	2.9	2.5	2.6	2.8	3	2.8	3.2	1.9
3	2	0.85	0	0.7	0.5	1.3	1.4	0.65	1.3	1.1	1.2	1.4	1.4	1.9	1.9	1.8	1.8	1.6	1.8	1.8	1.9	2.4	2	2.1	2.3	2.5	2.3	2.7	1.6
4	1.9	1.2	0.2	0	1.3	1.1	2	0.5	2	1.7	1.5	1.8	1.7	1.7	1.4	2.6	1.4	1.3	1.3	1.4	1.5	1.9	1.6	1.7	2.1	2	2.2	2.6	1.7
5	1.8	1.1	0.5	0.35	0	1.5	2	0.85	2	1.7	1.8	2.1	2	2.4	2.2	2.3	2.2	2	2.1	2.2	2.3	2.7	2.4	2.5	2.8	2.8	2.9	3.3	1.4
6	2.2	1.2	0.4	0.25	1.2	0	0.95	1.6	0.9	0.65	0.75	1	0.95	1.5	1.4	1.4	1.4	1.2	1.3	1.4	1.5	2	1.5	1.7	1.9	2.1	1.8	2.2	1.9
7	3.1	2.4	1.4	1.2	2.4	2.3	0	1.7	0.95	1	1.1	1.4	1.3	1.7	1.8	1.8	1.9	1.6	1.7	1.8	1.8	2.3	1.9	2.1	1.9	3.5	1.6	2	2.9
8	2.4	1.4	0.65	0.5	1.5	0.26	1.2	0	1.2	0.9	1	1.2	1.2	1.7	1.7	1.6	1.7	1.5	1.6	1.6	1.7	2.2	1.8	1.9	2.1	2.3	2.1	2.5	2.2
9	2.8	2.1	1.1	0.9	2.1	2	0.45	1.4	0	0.75	0.85	1.1	1	1.2	1.5	1.3	1.4	1.3	1.4	1.5	1.6	2	1.6	1.8	1.4	2.2	1.1	1.5	2.4
10	2.3	2.3	1.3	1.1	1.7	2.2	1.2	1.6	1.2	0	0.65	0.45	0.45	0.75	1.1	0.7	0.7	0.75	1	1	1	1.4	1.1	1.2	1.1	1.6	1.3	1.6	2
11	2.4	2	1	0.85	1.8	1.9	0.95	1.3	0.9	0.6	0	0.8	0.75	1.3	1.3	1	1.2	1	1.2	1.2	1.3	1.7	1.4	1.5	1.7	1.9	1.8	2.2	2.1
12	2	1.8	0.85	0.65	1.5	1.8	0.8	1.1	0.75	0.45	0.2	0	0.6	0.9	0.9	0.8	0.85	0.65	0.8	0.85	0.95	1.3	1	1.1	1.3	1.5	1.5	1.8	1.8
13	2.1	2.2	1.1	1	1.6	2.1	1.1	1.5	1	0.75	0.5	0.3	0	0.3	1	0.25	0.28	0.65	0.9	0.8	0.9	1.3	1	1.1	0.7	1.4	0.85	1.2	1.9
14	3.9	3.8	2.1	2	3.3	3.1	1.5	2.5	1.7	1.4	1.2	1	0.9	0	2.6	0.95	1	2.4	2.5	2.5	2.5	2.6	1.4	2.1	0.4	2.1	1.1	1.5	3.6
15	1.7	2.1	1.1	0.95	1.2	2	1.1	1.4	1	0.7	0.45	0.9	0.9	0.9	0	0.8	0.65	0.45	0.5	0.6	0.7	1.1	0.75	0.9	1.3	1.2	1.4	1.8	1.5
16	1.8	2.1	1.1	1	1.2	2	1.1	1.4	1	0.7	0.5	0.6	0.5	0.4	0.19	0	0.35	0.21	0.5	0.4	0.5	0.85	0.55	0.7	0.8	1	1	1.3	1.5
17	1.9	2.4	1.5	1.3	1.4	2.7	1.4	1.8	1.4	1	0.8	1	0.85	0.8	0.5	0.7	0	0.28	0.45	0.27	0.35	0.75	0.4	0.55	1.2	0.85	1.3	1.7	1.7
18	1.7	2.2	1.2	1	1.1	2.4	1.2	1.5	1.1	0.75	0.55	0.7	0.55	0.5	0.24	0.4	0.19	0	0.4	0.19	0.27	0.65	0.35	0.5	0.9	0.8	1	1.4	1.4
19	1.7	2.2	1.3	1.1	1.1	2.5	1.2	1.6	1.2	0.85	0.6	1	0.9	0.85	0.55	0.75	0.6	0.4	0	0.4	0.3	0.6	0.75	0.5	1.2	0.75	1.4	1.7	1.4
20	1.8	2.3	1.4	1.2	1.3	2.6	1.3	1.7	1.3	0.95	0.7	0.85	0.75	0.7	0.4	0.6	0.35	0.17	0.19	0	0.087	0.5	0.35	0.4	1.1	0.6	1.2	1.6	1.6
21	1.9	2.4	1.5	1.3	1.4	2.7	1.4	1.8	1.4	1	0.8	0.95	0.85	0.75	0.5	0.65	0.4	0.26	0.27	0.087	0	0.4	0.24	0.3	1.2	0.5	1.3	1.7	1.7
22	2	2.5	1.6	1.4	1.5	2.8	1.5	1.9	1.5	1.2	1	1.3	1.2	1.2	0.8	1.1	0.95	0.55	0.35	0.4	0.3	0	0.45	0.5	1.6	0.7	1.7	2.1	1.8
23	2	2.5	1.6	1.4	1.4	2.8	1.5	1.9	1.5	1.2	0.85	1	0.9	0.85	0.55	0.75	0.5	0.35	0.75	0.35	0.26	0.5	0	0.17	1.8	0.5	1.4	1.8	1.7
24	2.1	2.6	1.7	1.5	1.5	2.9	1.6	2	1.6	1.3	1	1.1	1	1	0.7	0.85	0.6	0.45	0.55	0.4	0.3	0.45	0.11	0	1.4	0.4	1.5	1.9	1.8
25	3.8	3.6	2	1.8	3.2	2.9	1.4	2.3	1.6	1.6	1.7	0.85	0.75	0.6	2.4	0.8	0.8	2.2	2.4	2.2	2.1	2.2	1.9	1.8	0	1.7	0.95	1.3	3.5
26	2.4	3	2	1.9	1.9	3.2	2	2.3	1.9	1.6	1.3	1.5	1.4	1.3	1	1.2	0.95	0.75	0.8	0.6	0.5	0.6	0.5	0.4	1.7	0	1.8	2.2	2.2
27	3.5	3.4	1.7	1.6	2.9	2.7	1.1	2	1.3	1.4	1.2	1	0.9	0.75	2.2	0.9	0.95	1.9	2.1	2.1	2.2	2.7	2.3	2.4	0.95	2.6	0	0.75	3.2
28	3.9	3.1	2.1	2	3.3	3	1.5	2.4	1.7	1.8	1.6	1.4	1.3	1.1	2.6	1.3	1.3	2.3	2.5	2.5	2.6	3.1	2.7	2.8	1.3	2.3	0.65	0	3.6
29	1.3	1.7	1	0.85	0.6	2	1.4	1.3	1.4	1.1	0.85	1.2	1.1	1	0.75	0.95	0.8	0.65	0.65	0.8	0.9	1.3	1	1.1	1.4	1.4	1.6	2	0

Table A5. Coordinates of the 147 locations.

Location	Latitude (Decimal Degrees)	Longitude (Decimal Degrees)
1	43.386866	−2.967695
2	43.385065	−2.969335
3	43.384389	−2.970417
4	43.385533	−2.970361
5	43.386875	−2.976296
6	43.385967	−2.975167
7	43.381338	−2.967009
8	43.382074	−2.969934
9	43.382585	−2.972301
10	43.381528	−2.972474
11	43.384161	−2.974454
12	43.383708	−2.975088
13	43.384144	−2.976702
14	43.384005	−2.976627
15	43.384120	−2.977955
16	43.383985	−2.977985
17	43.384236	−2.978082
18	43.383631	−2.978442
19	43.383199	−2.978524
20	43.383740	−2.979749
21	43.383519	−2.979418
22	43.383012	−2.979399
23	43.382807	−2.980273
24	43.383741	−2.980453
25	43.383798	−2.980842
26	43.383568	−2.980869
27	43.383510	−2.981054
28	43.382908	−2.980460
29	43.383270	−2.981680
30	43.382436	−2.981768
31	43.382340	−2.982082
32	43.382122	−2.982337
33	43.383011	−2.983951
34	43.382812	−2.983972
35	43.381533	−2.983377
36	43.381334	−2.983270
37	43.381829	−2.982122
38	43.382067	−2.986113
39	43.381568	−2.986365
40	43.381784	−2.988524
41	43.380237	−2.989935
42	43.381140	−2.989132
43	43.381249	−2.988139
44	43.381235	−2.989827
45	43.380718	−2.987577
46	43.380491	−2.987676
47	43.380686	−2.987247

Table A5. *Cont.*

Location	Latitude (Decimal Degrees)	Longitude (Decimal Degrees)
48	43.380298	−2.986529
49	43.379818	−2.985593
50	43.379931	−2.985285
51	43.380432	−2.984580
52	43.380354	−2.984325
53	43.380485	−2.983974
54	43.380450	−2.983486
55	43.379426	−2.987135
56	43.379123	−2.986105
57	43.378931	−2.986235
58	43.378462	−2.986664
59	43.380934	−2.982889
60	43.380474	−2.982763
61	43.380781	−2.981613
62	43.379854	−2.981971
63	43.380253	−2.982916
64	43.379868	−2.982576
65	43.379837	−2.982973
66	43.379389	−2.983617
67	43.379319	−2.983048
68	43.378805	−2.982968
69	43.378960	−2.982608
70	43.379314	−2.981559
71	43.381485	−2.980560
72	43.382740	−2.980696
73	43.380752	−2.980763
74	43.380498	−2.981023
75	43.380236	−2.980826
76	43.380501	−2.979872
77	43.381449	−2.979324
78	43.381796	−2.979530
79	43.382100	−2.978779
80	43.382583	−2.978996
81	43.382971	−2.979162
82	43.382396	−2.977782
83	43.382645	−2.977287
84	43.381542	−2.978895
85	43.381407	−2.977533
86	43.381706	−2.976436
87	43.382816	−2.978206
88	43.381285	−2.976771
89	43.381425	−2.975376
90	43.381242	−2.975296
91	43.380864	−2.975344
92	43.380848	−2.976503
93	43.380832	−2.977200
94	43.380186	−2.979504
95	43.380019	−2.980074
96	43.379958	−2.980587
97	43.378765	−2.981322
98	43.377759	−2.982722
99	43.377512	−2.982654
100	43.377344	−2.982785
101	43.381321	−2.992360
102	43.380362	−2.993327
103	43.380051	−2.993379
104	43.381045	−2.994875

Table A5. *Cont.*

Location	Latitude (Decimal Degrees)	Longitude (Decimal Degrees)
105	43.380561	−2.995728
106	43.379430	−2.993877
107	43.379093	−2.993990
108	43.379339	−2.992198
109	43.378773	−2.991203
110	43.378845	−2.990499
111	43.378604	−2.990441
112	43.378568	−2.990115
113	43.378646	−2.988672
114	43.378464	−2.987841
115	43.378462	−2.988852
116	43.378328	−2.987209
117	43.377763	−2.987006
118	43.377847	−2.986469
119	43.376388	−2.988905
120	43.376071	−2.988053
121	43.377174	−2.988653
122	43.377572	−2.988626
123	43.377533	−2.988873
124	43.377958	−2.988862
125	43.376747	−2.990108
126	43.375579	−2.989862
127	43.375428	−2.990021
128	43.375322	−2.990450
129	43.375529	−2.991228
130	43.374831	−2.991743
131	43.374468	−2.991131
132	43.373440	−2.991881
133	43.374343	−2.992137
134	43.378307	−2.991586
135	43.378739	−2.991224
136	43.378672	−2.990759
137	43.378758	−2.999239
138	43.378467	−2.990613
139	43.374214	−2.992418
140	43.374757	−2.993032
141	43.375189	−2.992432
142	43.375826	−2.994840
143	43.376289	−2.993976
144	43.377369	−2.992465
145	43.376535	−2.991381
146	43.376925	−2.991370
147	43.377435	−2.991360

The distances between the 147 locations can be found in [32].

References

1. European Commission. The Sustainable Development Goals—European Commission. Available online: https://ec.europa.eu/info/strategy/international-strategies/sustainable-development-goals_en (accessed on 7 February 2020).
2. Jovanovic, S. Expert Views (Part 2): What Are the Biggest Challenges in Municipal Waste Management in Serbia? Available online: https://balkangreenenergynews.com/expert-views-part-2-what-are-the-biggest-challenges-in-municipal-waste-management-serbia/ (accessed on 7 February 2020).
3. Caria, M.; Todde, G.; Pazzona, A. Modelling the Collection and Delivery of Sheep Milk: A Tool to Optimise the Logistics Costs of Cheese Factories. *Agriculture* **2018**, *8*, 5. [CrossRef]
4. Lin, S. Computer Solutions of the Traveling Salesman Problem. *Bell Syst. Tech. J.* **1965**, *44*, 2245–2269. [CrossRef]
5. Beltrami, E.J.; Bodin, L.D. Networks and vehicle routing for municipal waste collection. *Networks* **1974**, *4*, 65–94. [CrossRef]

6. Buhrkal, K.; Larsen, A.; Ropke, S. The Seventh International Conference on City Logistics The waste collection vehicle routing problem with time windows in a city logistics context peer-review under responsibility of the 7th International Conference on City Logistics. *Procedia Soc. Behav. Sci.* **2012**, *39*, 241–254. [CrossRef]
7. Tirkolaee, E.B.; Mahdavi, I.; Mehdi Seyyed Esfahani, M. A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time. *Waste Manag.* **2018**, *76*, 138–146. [CrossRef] [PubMed]
8. Hannan, M.A.; Akhtar, M.; Begum, R.A.; Basri, H.; Hussain, A.; Scavino, E. Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm. *Waste Manag.* **2018**, *71*, 31–41. [CrossRef] [PubMed]
9. De Bruecker, P.; Beliën, J.; De Boeck, L.; De Jaeger, S.; Demeulemeester, E. A model enhancement approach for optimizing the integrated shift scheduling and vehicle routing problem in waste collection. *Eur. J. Oper. Res.* **2018**, *266*, 278–290. [CrossRef]
10. Ramos, T.R.P.; de Morais, C.S.; Barbosa-Póvoa, A.P. The smart waste collection routing problem: Alternative operational management approaches. *Expert Syst. Appl.* **2018**, *103*, 146–158. [CrossRef]
11. Benjamin, A.M.; Beasley, J.E. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Comput. Oper. Res.* **2010**, *37*, 2270–2280. [CrossRef]
12. Cervera Paz, A.; Lopez Molina, L.; Rodriguez Cornejo, V. Removing the pillars of logistics: The physical Internet. *Dyna* **2018**, *93*, 370–375. [CrossRef]
13. Route4me. Why You Shouldn't Rely On Free Mapping Tools For Planning Routes. Available online: https://blog.route4me.com/2017/07/google-maps-planning-routes/ (accessed on 7 February 2020).
14. OptimoRoute. OptimoRoute—Route and Schedule Planning and Optimization For Delivery and Field Service. Available online: https://optimoroute.com/ (accessed on 7 February 2020).
15. Altexsoft. How to Solve Vehicle Routing Problems: Route Optimization Software and their APIs. Available online: https://www.altexsoft.com/blog/business/how-to-solve-vehicle-routing-problems-route-optimization-software-and-their-apis/ (accessed on 7 February 2020).
16. Goti, A.; Sanchez, A.; Oyarbide-Zubillaga, A. Money based maintenance model constrained multi-objective optimization. In Proceedings of the IADIS International Conference Applied Computing (IADIS 2006), San Sebastian, Spain, 25–28 February 2006; pp. 49–56.
17. Bac, F.Q.; Perov, V.L. New evolutionary genetic algorithms for NP-complete combinatorial optimization problems. *Biol. Cybern.* **1993**, *69*, 229–234. [CrossRef]
18. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
19. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Boston, FL, USA, 1989.
20. Davis, L. Applying adaptive algorithms to epistatic domains. *Proc. Int. Jt. Conf. Artif. Intell.* **1985**, *85*, 162–164.
21. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer: Berlin, Germany, 1992.
22. Reeves, C. *Modern Heuristic Techniques for Combinatorial Problems*; Blackwell Scientific Publications: Oxford, UK, 1993.
23. Larrañaga, P.; Kuijpers, C.; Murga, R.; Inza, I.; Dizdarevich, S. Evolutionary algorithms for the traveling salesman problem: A review of representations and operators. *Artif. Intell. Rev.* **1999**, *13*, 129–170. [CrossRef]
24. Malhotra, R.; Chug, A. Software Maintainability: Systematic Literature Review and Current Trends. *Int. J. Softw. Eng. Knowl. Eng.* **2016**, *26*, 1221–1253. [CrossRef]
25. Europa Press. Spain Needs 10,000 Programmers Due to The Lack of Computer Engineers, According to the JBCNConf (España Necesita 10.000 Programadores por la Falta de Ingenieros InformáTicos, Según El JBCNConf). *La Vanguard*, 26 March 2019.
26. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
27. Lin, S.; Kernighan, B.W. An effective heuristic algorithm for the Traveling Salesman Problem. *Oper. Res.* **1973**, *21*, 498–516. [CrossRef]
28. Ambati, B.K.; Ambati, J.; Mokhtar, M.M. Heuristic Combinatorial Optimization by Simulated Darwinian Evolution: A Polynomial Time Algorithm for the Traveling Salesman Problem. *Biol. Cybern.* **1991**, *65*, 31–35. [CrossRef]

29. De Jong, K.; Spears, W.M. Using Genetic Algorithms to Solve NP Complete Problems. In Proceedings of the Third International Conference on Genetic Algorithm, Los Altos, CA, USA, 4–7 June 1989; pp. 124–132.
30. Huella Medioambiental (Environmental Footprint)—Eco Calculator. Available online: https://ecocalculator.renault-trucks.com/es/huella-medioambiental (accessed on 8 January 2020).
31. Observatorio de Costes del Transporte de Mercancías por Carretera (Observatory of Road Transport Freight Costs). Available online: https://www.fomento.gob.es/CVP/handlers/pdfhandler.ashx?idpub=TTW103 (accessed on 7 April 2020).
32. Distances of the 147 Locations. Available online: https://docs.google.com/spreadsheets/d/1ZkiBnn-tGtyWV6pdxGEnqTBKIlPHJO3JqHkV5K5Nb6E/edit?usp=sharing (accessed on 8 January 2020).