Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 18(6): 9579–9605.
DOI: 10.3934/mbe.2021470
Received: 11 August 2021
Accepted: 12 October 2021
Published: 03 November 2021

Research article

Routing in waste collection: A simulated annealing algorithm for an Argentinean case study

Diego G. Rossit[1,2,*], Adrián A. Toncovich[1] and Matías Fermani[1]

[1] Department of Engineering, Universidad Nacional del Sur, Alem Av. 1253, Bahía Blanca 8000, Argentina
[2] INMABB UNS-CONICET, Alem Av. 1253, Bahía Blanca 8000, Argentina

* **Correspondence:** Email: diego.rossit@uns.edu.ar; Tel: +542914595000 ext: 3233; Fax: +542914595157.

Abstract: The management of the collection of Municipal Solid Waste is a complex task for local governments since it consumes a large portion of their budgets. Thus, the use of computer-aided tools to support decision-making can contribute to improve the efficiency of the system and reduce the associated costs, especially in developing countries, which usually suffer from a shortage of resources. In the present work, a simulated annealing algorithm is proposed to address the problem of designing the routes of waste collection vehicles. The proposed algorithm is compared to a commercial solver based on a mixed-integer programming formulation and two other metaheuristic algorithms, i.e., a state-of-the-art large neighborhood search and a genetic algorithm. The evaluation is carried out on both a well-known benchmark from the literature and real instances of the Argentinean city of Bahía Blanca. The proposed algorithm was able to solve all the instances, having a performance similar to the large neighborhood procedure, while the genetic algorithm showed the worst results. The simulated annealing algorithm was also able to improve the solutions of the solver in many instances of the real dataset.

Keywords: municipal solid waste; waste collection; vehicle routing problem; simulated annealing; large neighborhood search; genetic algorithm; mixed-integer programming

1. Introduction

The generation of municipal/urban solid waste (MSW) is an inalienable consequence of the development of modern cities. Likewise, its correct management is one of the key elements for the sustainable development of a city [1]. The stages in the reverse logistics chain of MSW are diverse. According to Tchobanoglous et al. [2], they can be classified into waste generation; handling, separation, accumulation and processing of waste at source; collection, transfer and transportation; separation, processing and transformation of solid waste; and finally, disposition. In these stages, there are many decisions to be made. For this reason, proposing computer-aided tools that allow assisting decision-making agents can contribute to a more efficient use of resources.

This work focuses on the collection, transfer and transportation stage of waste, which is known to be among the most expensive activities of MSW systems [3]. This activity comprises the design of the waste collection routes for the vehicles that collect the waste produced by households and other commercial and institutional urban activities. The correct performance of this task is critical for due quality of service provided to the citizens since when it is mishandled it can lead to waste bins overflow which is associated with environmental and social issues. In particular, a simulated annealing (SA) metaheuristic algorithm is proposed to solve this problem. The proposed algorithm is compared against other metaheuristic algorithms and CPLEX in order to study its performance. The experimentation is carried out on both a well-known benchmark of the literature and real instances of the city of Bahía Blanca, Argentina. This paper is a revised and expanded version of our conference work presented at The Xth international conference of production research-Americas 2020 [4].

This article is structured as follows. Section 2 presents the proposed model and resolution algorithms for the addressed problem and a literature review of the main related works. Section 3 describes the results of the computational experimentation performed on a well-known benchmark set of instances from the literature and a real-world case of study. Section 4 analyses and discusses the obtained results. Finally, Section 5 presents the main conclusion and future research work lines.

2. Materials and methods

The waste collection problem can be addressed considering multiple features depending on the collection system and the real-world restrictions that are taken into account. The waste collection problem that is addressed in this work is based on a community bin system in which waste is carried by generators from their households to the collection points that are distributed throughout the city. A collection point is a place in the city specially conditioned for installing a (waste) bin or set of bins.

2.1. Mathematical formulation

This problem can be modelled either as a Capacitated Vehicle Routing Problem (CVRP) or a soft clustered-CVRP, in which the collection points are split into regions (clusters) and within a route a vehicle visiting a collection point in a cluster must visit all the remaining collection points therein before leaving it or to respect the soft-cluster constraint, i.e., all collection points of the same cluster must be served by the same vehicle [5]. In this work, the waste collection problem is modelled as a CVRP in which a fleet of homogeneous vehicles has to visit all the collection points considering capacity constraints. Given a set of collection points C and a superset $\underline{C} = C \cup c_0$, where c_0 is the depot where the vehicles start and end their trips, the following variables and parameters are defined: binary variable x_{ij} that takes value 1 if a vehicle uses the path from collection point $i \in C$ to collection point $j \in C$ and 0 otherwise; the continuous non-negative auxiliary variable u_i for subtour elimination indexed over collection points ($i \in C$); parameter Q is defined as the capacity of the vehicle; parameter d_{ij} as the distance between collection points $i \in C$ and $j \in C$; and parameter q_i as the amount of waste to be collected at collection point $i \in C$. In this way, the following model is proposed, using the two-index formulation developed by Miller et al. [6] in Eqs (1)–(6).

$$\min D = \sum_{i,j \in \underline{C}} x_{ij} d_{ij} \tag{1}$$

Subject to:

$$\sum_{\substack{j \in \underline{C} \\ j \neq i}} x_{ij} = 1, \forall \ i \in C \tag{2}$$

$$\sum_{\substack{j \in \underline{C} \\ j \neq i}} x_{ji} = 1, \forall \ i \in C \tag{3}$$

$$u_i - u_j \leq Q(1 - x_{ij}) - q_j, \forall \ i, j \in C, i \neq j \tag{4}$$

$$q_i \leq u_i \leq Q, \forall \ i \in C \tag{5}$$

$$x \in \{0,1\} \tag{6}$$

The proposed objective is to minimize the total distance travelled expressed in Eq (1). Equations (2) and (3) guarantee that each collection point is visited only once, having a single successor collection point and a single predecessor collection point on the route. Equation (4) prevents the formation of subtours. Equation (5) ensures that the vehicle's capacity is not exceeded. Equation (6) establishes the binary nature of the corresponding variable.

2.2. Proposed resolution method: a simulated annealing algorithm

As it was previously said, the collection of MSW in bins constitutes an application case of the VRP (Vehicle Routing Problem) problem. In computational complexity theory, this type of problem is classified as NP-hard [7], that is, it is at most as difficult as problems for which efficient solving algorithms that run in polynomial time have not yet been developed, regarding the size of the problem [8]. In this type of problems, metaheuristic tools allow obtaining good solutions in reasonable computational times [9]. The proposed resolution algorithm is based on a simulated annealing (SA) method and was adapted from the previous algorithm developed in Toncovich et al. [10,11]. SA is a local search-based method that was developed from an analogy with the physical phenomenon of annealing [12] to solve complex optimization problems. Local search methods search for the solution with the best value of the chosen criteria in the current solution neighborhood, accept it as the current solution, and repeat this procedure until it is not possible to improve the solution in the explored neighborhood. By systematically applying this procedure, a local optimum for the problem is generally obtained. To avoid being trapped in a local optimum, a diversification mechanism must be incorporated in order to adequately explore the solutions space. In simulated annealing metaheuristics, the diversification strategy allows moves, with some probability, toward solutions that worsen the current value of the objective function.

To get a good approximation to the optimal solution of the problem during the search process, it is necessary to restart the search regularly from one of the solutions accepted during the search process selected at random. The SA algorithm incorporates the classic parameters of simulated annealing. The parameters and variables corresponding to the implemented SA algorithm are indicated below.

- t: current iteration.
- n: current step.
- S_0: initial solution.
- S_A: current solution.
- S_c: candidate solution.
- S^*: best solution found.
- $V(S)$: neighborhood for solution S, given by the set of solutions that can be obtained from solution S through a basic perturbation. In our implementation the perturbation is generated by swapping two randomly selected elements (collection points) in S.
- T: control parameter. This variable simulates the temperature in the real annealing process. It is a positive real number that varies in the interval $T \in [T_f, T_0]$ during the execution of the algorithm, where T_0 is the initial temperature and $T_0 > T_f$.
- N_T: maximum number of iterations performed by the algorithm for a certain temperature value T_n, at step n.
- $\alpha(T)$: function that determines the cooling mechanism, i.e., how T varies during the search process. In this case, it is a geometric progression of the form $T_n = \alpha \times T_{n-1}$, with $\alpha \in [0{,}8; 0{,}999]$.
- N_{cnt}: number of consecutive iterations without improvement in the objective function value at the iteration t.
- N_{stop}: maximum allowable number of iterations without improvement.
- $time\_elapsed$: time that has passed since the algorithm started.
- $time\_limit$: amount of time available for executing the algorithm on a given instance.
- p_A: probability of accepting a solution that is worse than the current solution.
- ε: a uniformly distributed random number in the interval (0,1).

The pseudo-code of the SA algorithm is presented in Algorithm 1.

Algorithm 1: Simulated annealing algorithm.

```
Inputs: data of the instance ($d_{ij}$, $q_i$, $Q$) and parameters $T_0$, $T_f$, α, $N_T$, $N_{stop}$, $time\_limit$
Initialization
        $time\_elapsed \leftarrow 0$; $time \leftarrow now()$
        create $S_0$ using a randomized greedy procedure
        evaluate $S_0$, $D(S_0)$, using Equation (1)
        $S_A \leftarrow S_0$; $S^* \leftarrow S_0$; $t \leftarrow 1$;   $n \leftarrow 1$; $N_{cnt} \leftarrow 0$; $T_1 \leftarrow T_0$
repeat
        $counter \leftarrow 0$
        repeat
            create a new solution $S_c$, $S_c \in V(S_A)$
            evaluate $S_c$, $D(S_c)$
            $Cost \leftarrow D(S_c) - D(S_A)$
            if $Cost \leq 0$ then
                $S_A \leftarrow S_c$;  ; $N_{cont} \leftarrow 0$
                if $D(S^*) < D(S_A)$ then
                    $S^* \leftarrow S_A$
                endif
            else
                $p_A \leftarrow e^{-\frac{Cost}{T_n}}$; $\varepsilon \leftarrow rand(0,1)$
                if $\varepsilon \leq p_A$ then
                    $S_A \leftarrow S_c$; $N_{cnt} \leftarrow 0$
                else
                    $N_{cnt} \leftarrow N_{cnt} + 1$
                endif
            endif
            $t \leftarrow t + 1$; $time\_elapsed \leftarrow now() - time$; $counter \leftarrow counter + 1$
        until $counter \geq N_T$
        $n \leftarrow n + 1$; $T_n \leftarrow \alpha \times T_{n-1}$
until $N_{cnt} > N_{stop}$ OR $T_n > T_f$ OR $time_{elapsed} > time\_limit$
return $S^*$
```

2.3. Comparison algorithms

The proposed metaheuristic is compared with a commercial solver and other two metaheuristics, i.e., large neighborhood search (LNS) algorithm developed by Erdoğan [13] and a standard genetic algorithm.

2.3.1. Commercial solver based on MIP formulation

The solver that is used to solve the proposed mathematical formulation presented in Section 2.1 is CPLEX. In order to speed up the resolution process, the Eq (7) is added that sets the minimum number of trips required considering the overall waste to collect.

$$\sum_{j \in C} x_{c_0 j} \geq \sum_{j \in C} \frac{q_i}{Q} \tag{7}$$

This Equation is a well-known valid inequality or cut in routing problems [14]. In preliminary tests, this cut was found to be competitive for reducing the computing times. This is generally associated to a better estimation of the lower bound due to a tighter approximation of the feasible region of the linear relaxation to the feasible region of the actual MIP model in resolution algorithms based on branch-and-cut [15] as CPLEX.

2.3.2. Large neighbourhood search

Erdoğan’s metaheuristic was adapted from the adaptive LNS heuristic developed by Pisinger and

Ropke [16], which extends Shaw's original heuristic [17]. A simplified explanation of its operation and pseudo-code is presented in Algorithm 2.

The operation of the heuristic consists of two main stages. The first is the construction of an initial solution that is obtained by inserting clients into the available routes in accordance with the objective to be minimized (operator $InitialSol()$). Then, an enhancement is made using the $LocalSearch()$ operator that applies four local search techniques. The first three are known as Exchange (two clients are exchanged), 1-OPT (a client is extracted from one route and reinserted into another) and 2-OPT (two route segments are exchanged). These operators are detailed in Groër et al. [18]. The fourth local search operator is Vehicle-Exchange. This operator tries to exchange the vehicles assigned to two routes, which is why it is useful when working with a heterogeneous fleet.

The second stage is where you try to get an improvement on the solution from the previous stage. The $DestroyAndRebuild()$ operator is applied, which considerably alters the current solution to explore another region of the search space and thus be able to escape from local optimum. In particular, long sections of routes are extracted from the current solution and the extracted clients are reinserted in those positions that minimize the objective function (function $D()$ defined in Eq (1)). A solution is accepted if it has a better cost than the best solution found so far or, if a solution does not have a better cost than the previous solution, it can even be accepted with a probability p-value. This process is repeated cyclically up to the time limit imposed by the user.

Algorithm 2: Pseudo-code of large neighborhood search [13].

```
**Input:** data of the instance ($d_{ij}$, $q_i$, $Q$) and parameter $Time\ \ limit$
**Initialization**
$S_0 \leftarrow InitialSol()$
$S' \leftarrow LocalSearch()$
$BestSol \leftarrow S'$
$Time\ \ elapsed \leftarrow 0$
**repeat**
    $time \leftarrow now()$
    $S' \leftarrow DestryAndRebuild(S')$
    $S' \leftarrow LocalSearch(S')$
    **if** $D(S') \leq D(BestSol)$
        $BestSol \leftarrow S'$
    **else**
        $\varepsilon \leftarrow rand(0,1)$
        If $\varepsilon \leq pvalue$
            $BestSol \leftarrow S'$
        **end if**
    **end if**
    $Time\ \ elapsed \leftarrow now() - time$
**until** $Time\ \ elapsed > Time\ \ limit$
**return** $BestSol$
```

2.3.3. Genetic algorithm

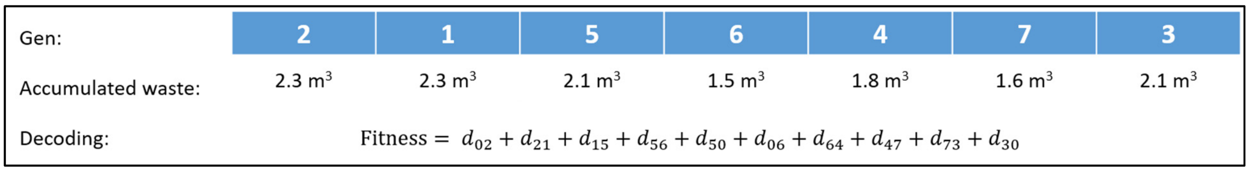
The schema of the implemented genetic algorithm (GA) is presented in in Algorithm 3.

Algorithm 3: Pseudo-code of genetic algorithm [19].

```
Input: data of the instance ($d_{ij}$, $q_i$, $Q$) and parameters $p_c$, $p_m$, $\#P$, and $Time\ \ limit$
Initialization
$P \leftarrow InitialPopulation()$
$Time\ \ elapsed \leftarrow 0$
repeat
    $time \leftarrow now()$
    $evaluate(P)$
    $parents\ \ \leftarrow selection(P_t)$
    $offspring\ \ \leftarrow \mathbf{crossover}(parents)$
    $offspring\ \ \leftarrow \mathbf{mutation}(offspring)$
    $P \leftarrow \mathbf{variation\ \ operations}(parents)$
until $Time\ \ elapsed > Time\ \ limit$
return $best\ \ individual\ \ found$
```

The GA was developed using the Distributed Evolutionary Algorithms in Python (DEAP) framework [20] and has the following characteristics:

- *Representation of solutions*. Solutions are encoded as a permutation of integers of length equal to the number of containers n. Each index in the vector represents the visit order in the tour and the corresponding integer value represents one of the containers.
- *Initialization*. The population of size $|P|$ is initialized by applying a random procedure to generate the permutations with a uniform distribution.
- *Genetic operators*. The recombination operator is the Partially Mapped Crossover (PMX) applied on two selected individuals with p_c probability. The mutation operator is based on Swap Mutation and swaps two elements of the permutation. Both selected operators have been usually applied in CVRP problems [21]. The mutation applies to an individual with probability pm. The proposed operators guarantee the feasibility of the solution.
- *Selection*. The selection is made through a binary tournament, from which the one with the best fitness is selected.
- *Replacement*. In each iteration, the new population is made up of the best 10% (with better fitness) of the previous population and the rest of new individuals generated through genetic operators.
- *Fitness assessment*. The fitness function is decoded by reading alleles in the gen from left to right and inserting a depot visit when necessary due to the capacity constraint of the vehicle. An example of the decoding process is presented in Figure 1 for a representative gen considering a vehicle with a capacity of $7m^3$. The collection vehicle starts its route from the depot and collects the waste of GAPs 2,1 and 5. However, collecting the waste of GAP 6 is not feasible since the amount of waste ($1.5m^3$) exceeds the available remaining capacity of the vehicle, i.e., $7\ m^3$ - $2.3\ m^3 - 2.3\ m^3 - 2.1\ m^3 = 0{,}3\ m^3$. Thus, a visit to the depot to unload the vehicle is inserted before visiting GAP 6, adding the distances d_{50} and d_{06} to the fitness. Then, the vehicle visits GAPs 4,7 and 3 and, finally, it returns to the depot.



Gen:	2	1	5	6	4	7	3
Accumulated waste:	2.3 m^3	2.3 m^3	2.1 m^3	1.5 m^3	1.8 m^3	1.6 m^3	2.1 m^3
Decoding:	Fitness = $d_{02} + d_{21} + d_{15} + d_{56} + d_{50} + d_{06} + d_{64} + d_{47} + d_{73} + d_{30}$						

Figure 1. Example of fitness assessment.

- *Cut due to stagnation*. A deadlock cut-off mechanism was incorporated. The algorithm stops when no improvement in the fitness of the best individual is obtained during a time interval equal to 50% of the maximum execution time of the instance. Otherwise, it continues until the generation—evaluation, selection, crossing, mutation and replacement cycle—which has exceeded the maximum execution time at the end.

2.4. Related works

The problem of collection of containerized waste in a city has been frequently addressed in the literature. In Beliën et al. [22] and Han and Ponce Cueto [23], extensive reviews of these works are developed.

Among the main works that applied heuristics in waste collection, it is the work of Kim et al. [24] who applied a clustering-based heuristic and an insertion heuristic on a set of real-world instances of the United States of America. Over the same dataset, Benjamin and Beasley [25] applied two advanced metaheuristics, i.e., a variable neighborhood search and a tabu search, improving the previous results of Kim et al. [24]. Nesmachnow et al. [9] proposed an evolutionary algorithm to optimized collection routes in Montevideo, Uruguay while considering priorities in collection points that are located near public institutions or busy places in which overflowing may have a large social impact.

Regarding SA algorithms in waste management, there is the work of Tirkolaee et al. [26]. The authors applied a metaheuristic based on SA improved with an efficient cooling equation to address the problem of waste collection considering uncertainty in waste generation that aims at minimizing the total distance and the usage cost of vehicles. The proposed algorithm obtained similar results than the exact solution for small instances. A comprehensive work was performed by Nowakowski et al. [27] who compared four different heuristics, i.e., simulated annealing, tabu search, greedy, and bee colony optimization, to solve case studies of three different cities in the context of electronic waste collection. The results showed that SA outperforms the rest of the proposed heuristics. A similar problem is addressed in Tirkolaee et al. [3] in which more priority is assigned to collect waste bins that are located near some special places, such as health centers that produce hazardous waste, than to household waste for addressing a case study in Sanandaj, Iran. The proposed SA-based algorithm was able to obtain near-optimal solutions in comparison with the CPLEX solver. Mekamcha et al. [28] proposed a SA to deal with a waste collection case study in the Algerian city of Tlemcen in which the collection is modelled as a Travelling Salesman Problem. The SA algorithm is compared with a Tabu Search obtaining better results regarding total distance minimization.

In the case of Argentina, some studies can be found on applications of VRP models to solve the problem of design of collection routes. For example, Bonomo et al. [29] implemented a solution strategy based on integer programming to schedule the collection routes for waste containers in the Southern area of the City of Buenos Aires. On the other hand, in Bonomo et al. [30] a different model is presented for this city, which aims at minimizing the travel distances simultaneously with minimizing wear and tear on vehicles. In the city of Concordia, Bertero [31] presents an application to design the city's collection routes, making an effort to minimize the number of turns to facilitate the implementation of the routes by the authorities. On the other hand, Bianchetti et al. [32] exhibits an algorithm to solve the zoning of the city of San Miguel de Tucumán in order to optimize the use of resources, reassigning trucks to the downtown area of the city. In Braier et al. [33], through mathematical programming models, the collection of recyclable waste is planned for the city of Morón. Delle Donne et al. [34] presented a solution strategy based on integer programming to scheduling of routes to collect waste produced by leaf sweeping of streets in the city of Trenque Lauquen.

From the analysis of the related literature, we consider that there is still room to propose efficient algorithms based on simulated annealing to design waste collection routes. Moreover, this work compares SA with other metaheuristic approaches and a commercial solver in order to assess the competitiveness of the simulated annealing procedure and addresses a real case study of the city of Bahía Blanca.

3. Results of computational experimentation

This Section presents the set of benchmark instances from the literature, the real-world case from the city of Bahía Blanca, the implementation details for the solution algorithms, the obtained results and an analysis of these results.

3.1. Case study: Argentinean city of Bahía Blanca

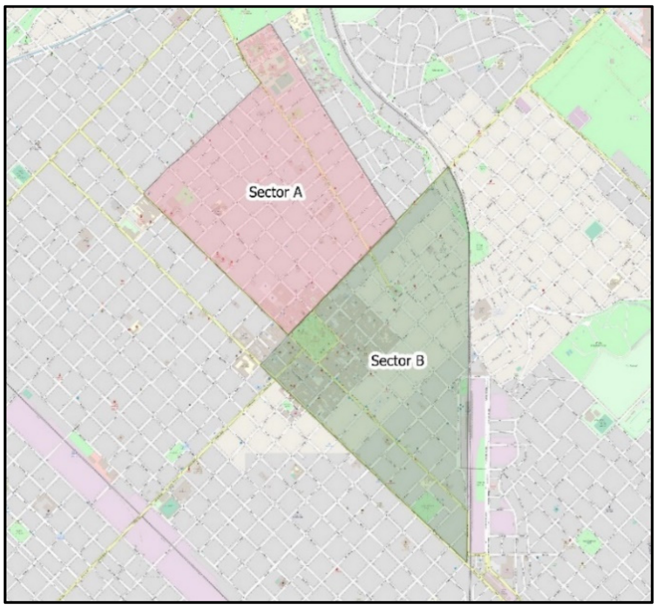


Figure 2. Case study of the city of Bahía Blanca.

Bahía Blanca is a medium-size city located in the South of Argentina: It is considered as a major industrial, logistic and university center from the southern part of the country. The management of urban solid waste in the city of Bahía Blanca is a crucial activity for local authorities, not only because of the broad environmental and social impact associated with its operation but also because it consumes a large part of the municipality's budgetary resources [35]. Therefore, approaches that allow the optimization of collection logistics may be relevant to achieve a more efficient service provision [8,36]. Nowadays, waste collection is performed in a door-to-door basis. However, there are initiatives and studies that aim at migrating to a community bins system [8,35–38]. This work uses a real dataset that was obtained in one of these projects carried out by the academic, municipality and private stakeholders of the city [35]. This dataset was built with the aim of migrating from the current door-to-door waste collection system to the community waste bins system and it corresponds to two neighborhoods of the downtown of Bahía Blanca. These neighborhoods can be depicted in Figure 2: The University neighborhood as sector A and the downtown as sector B.

Table 1. Real MSW instances description.

Instance	Sector	Type of waste	Collection frequency	Number of collection points	Waste generation per collection point (m^3)	
					Average	Std. dev.
A-F2HUM	A	Humid	6 per week	69	0.83	0.22
A-F2DRY	A	Dry	4 per week	69	2.22	0.62
A-F3HUM	A	Humid	6 per week	79	0.73	0.18
A-F3 DRY	A	Dry	3 per week	79	2.94	0.66
B-F2HUM	B	Humid	6 per week	68	0.79	0.21
B-F2 DRY	B	Dry	4 per week	68	2.85	0.79
B-F3HUM	B	Humid	6 per week	101	0.55	0.17
B-F3 DRY	B	Dry	3 per week	101	2.95	0.69
A + B-F2HUM	A + B	Humid	6 per week	137	0.81	0.21
A + B-F2 DRY	A + B	Dry	4 per week	137	2.55	0.74
A + B-F3HUM	A + B	Humid	6 per week	180	0.63	0.19
A + B-F3DRY	A + B	Dry	3 per week	180	2.95	0.67

One of the goals of the aforementioned project [35] was also to implement source classification of waste to increase the amount of waste that is recycled in the city and assess the impact of different collection frequencies in the number of collection points. Following these guidelines for this work, the real instances are defined as follows. Firstly, three different geographic areas are considered: Sector A, Sector B, and the overall Sector A + B. From each area, four different instances are constructed, which differ in the type of waste -humid or dry waste- and the collection frequency considered. The instances of humid waste are always collected six times per week since they cannot remain uncollected due to environmental reasons. However, dry waste can be collected either four times per week or three times per week. Thus, Cavallin et al. [35] proposed two systems based on collection frequency for the city: The F2 system in which humid waste is collected six times per week and dry waste is collected four times per week, and F3 system in which dry waste is collected six times per week and dry waste is collected three times per week. Considering the three factors, i.e., geographic sector, type of waste and collection frequency in the system, the twelve instances of Table 1 are defined[1]. Further details of the instances can be consulted in Cavallin et al. [35]. In regard to the present work, the collection

[1] The instances of Bahía Blanca used in this paper can be consulted in GitHub: https://github.com/diegorossit/Instances-of-Waste-collection-in-Bah-a-Blanca/blob/afdc3cef822ef7c49c0f166e4360136e9c7cf69e/README.md.

frequency of the dry waste will affect the stored quantity of dry waste to be collected, and due to its influence on the required storage space in the collection point, it will also affect the quantity of humid waste to be collected.

The capacity of the truck was set in 21 m^3 which is the common capacity in the trucks of the fleet of the collection company. The distances between collection points were estimated with Open Source Routing Machine[2] using the approach proposed by Vázquez [39].

3.2. Benchmark instances

In regard to the CVRP benchmark instances, the well-known set E of 13 instances proposed in Christofides and Eilon [40] was used. Both the instances and the optimal solutions were retrieved from the VRP-REP digital repository [41].

3.3. Implementation details

For solving the MIP formulation of Eqs (1)–(7), the commercial solver IBM ILOG CPLEX 20.1 was used [42]. It was implemented using GAMS 35.1 [43] as a modelling language. The majority of the parameters were kept in their default values [42,43] except for the enlargement of the working memory from the default 2048 MB to 5000 MB. This parameter was changed since on preliminary experimentation the solver ran out of memory space using the default configuration. The model that was solved with CPLEX is the MIP model composed by Eqs (1)–(7), which includes the valid inequality of the minimum number of trips (Eq (7)) to accelerate convergence. A time limit of 15,000 seconds was set to CPLEX for each run. The exact resolution was only applied to the real-world scenarios since the optimal solution of the benchmark instances is already available.

The SA was programmed in Visual Basic for Applications (VBA), the same programming language used in the LNS implementation taken from the repository of the author[3]. The GA algorithm was implemented using the DEAP library as the programming language.

Regarding the calibration of the metaheuristics the LNS does not need calibration [13]. The SA algorithm was fine-tuned using a small number of test runs in order to establish the values for the parameters specified in Section 2.2.

The GA algorithm was calibrated with a parametric analysis. The parameters that were analyzed through a statistical analysis were the size of the population $|P|$—the values 100; 150; and 200 were considered, the probability of mutation p_m—the values 0.05; 0.1; 0.15; 0.20; and 0.25 were considered and crossover p_c—the values 0.5; 0.6; 0.7; 0.8; 0.9; and 1 were tested. For carrying out the parametric analysis the instances P-n16-k8, P-n19-k2 and P-n20-k2 proposed in [44] were used. For each instance and for each parametric configuration, 50 independent executions were carried out. The Shapiro-Wilk test was applied to study whether the fitness distribution had a normal distribution. Since many executions did not fit a normal distribution, the medians were evaluated with the Friedman test [45], selecting the configuration: $|P| = 200$, $p_c = 0.9$ and $p_m = 0.25$ as the most promising parametric configuration.

In order to provide a fair comparison between different metaheuristics the same computing time limits and implementation computer should be used [46]. In this case, the three metaheuristic were run

[2] http://project-osrm.org/.

[3] https://people.bath.ac.uk/ge277/vrp-spreadsheet-solver/.

in a personal computer with an Intel Core i5-3570k@3.40 GHz processor and 12 GB of RAM memory on a Windows 10 environment. For solving the instances, 30 runs were performed with each metaheuristic to obtain a more representative sample of the performance of the algorithms. Regarding execution times, Equation (8) was used to set the maximum execution time of each run in seconds (M) for the metaheuristics which is proportional to the number of collection points in the instance - excluding the depot-(N)

$$M = \frac{(N \quad \times 10 \times 60)}{50} \tag{8}$$

It should be taken into account that both the SA and the GA described here have cut-off mechanisms due to stagnation that can lead to a real execution time that is less than the maximum. This is not the case for the LNS whose execution time will be equal to the maximum. In the case of CPLEX, a time limit of 15,000 seconds was given to the solver.

3.4. Computational experimentation

In this Section, we present the results of the computational experimentation. The experimentation was performed over two different datasets: the real world MSW instances of Bahía Blanca described in Section 3.1 and the benchmark instances described in Section 3.2, respectively.

Since the three metaheuristics involve probabilistic parameters, the results were statistically analyzed according to the following procedure. Each metaheuristic was run 30 times on each instance. Then, a two-stage statistical methodology is applied. First, the Shapiro-Wilk normality test [47] is used to determine whether the results follow a normal distribution. The null hypothesis of this test is that the data has a normal distribution and the alternative hypothesis is that the data cannot be assured to have a normal distribution. The test was applied with a statistical confidence level of 0.95.

Table 2. Results of the real instances of the three metaheuristics.

Instance	SA				LNS				GA			
	min	median	max	iqr	min	median	max	iqr	min	median	max	iqr
A-F2HUM	25.99	***26.81**	27.91	0.75	**25.88**	*26.87	27.90	0.73	32.80	36.59	41.79	2.95
A-F2DRY	**46.13**	***46.89**	47.41	0.25	46.41	***46.89**	48.20	0.33	52.93	55.67	58.86	1.75
A-F3HUM	**27.31**	**27.95**	29.30	0.39	27.73	28.81	29.33	0.66	37.48	40.74	43.23	2.50
A-F3DRY	**66.23**	66.74	67.15	0.39	65.95	**66.56**	67.30	0.38	72.61	75.36	77.59	2.20
B-F2HUM	**24.56**	**24.96**	26.02	0.52	24.72	25.30	25.60	0.03	31.01	34.24	36.03	3.24
B-F2DRY	39.80	*40.41	42.88	0.35	**39.65**	***40.36**	40.90	0.48	45.72	48.13	52.64	2.76
B-F3HUM	**29.04**	**29.54**	30.44	0.73	29.30	30.16	31.20	1.12	44.12	47.59	53.63	2.75
B-F3DRY	**57.35**	*58.33	58.89	0.44	57.36	***58.32**	58.97	0.36	68.86	72.66	77.03	3.75
A + B-F2HUM	**48.43**	**49.30**	50.54	0.88	50.14	50.88	51.83	0.89	77.82	82.91	88.79	5.54
A + B-F2DRY	88.99	**90.51**	92.51	1.11	**88.54**	91.59	94.61	1.98	111.65	115.34	121.69	3.30
A + B-F3HUM	**54.83**	*56.54	59.19	1.37	55.10	***56.25**	57.78	0.78	114.27	127.10	140.38	5.75
A + B-F3DRY	**121.24**	**122.72**	123.96	1.10	125.87	128.70	131.30	1.56	174.92	182.04	186.23	5.75

After that, in case the results of the three metaheuristics for a specific instance follow a normal distribution, the mean value is used as estimator for the studied metric and the standard deviation (std) is used as a measure of statistical dispersion, and the parametric ANOVA statistical test is applied to analyze if there are significant differences on the mean values of the metaheuristics. Conversely, in case the results of any of the metaheuristics does not follow a normal distribution, the median value is used as estimator for the studied metric and the interquartile range (IQR) as a measure of statistical dispersion, and the non-parametric Kruskal-Wallis statistical test [48] is applied to analyze if there are significant differences on the median values of the metaheuristics. The null hypothesis of this test is that the medians are equal and the alternative hypothesis is that the medians are not equal. It was applied with a statistical confidence level of 0.95.

In both set of instances, at least the results of one of the metaheuristics rejected the hypothesis of normal distribution according to Shapiro Wilk test. Thus, medians and interquartile ranges are used to report the results and the Kruskal-Wallis test is used to analyze the differences among medians. The detailed results of the Shapiro Wilk and Kruskal-Wallis tests can be consulted in Appendix.

For the real instances Table 2 reports for each metaheuristic: the minimal value, the median value, the maximal value, and the interquartile range/standard deviation of the D values achieved by each metaheuristic. In the pairwise comparisons performed by Kruskal-Wallis statistical test all the medians where found to be statistically different except for those that have an asterisk*, which rejected this hypothesis.

Table 3. Percentage difference of the solutions of each metaheuristic to the exact solution.

Instance	CPLEX		Comparison with minimal $\boldsymbol{D}$			Comparison with median $\boldsymbol{D}$		
	D	CPLEX gap %	Δ^{SA}	Δ^{LNS}	Δ^{GA}	Δ^{SA}	Δ^{LNS}	Δ^{GA}
A-F2HUM	25.49	*0.00%	1.96%	1.53%	28.63%	5.18%	5.41%	43.55%
A-F2DRY	47.18	10.73%	–2.23%	–1.63%	10.49%	–0.61%	–0.61%	17.99%
A-F3HUM	27.21	*0.00%	0.37%	1.91%	36.49%	2.72%	5.88%	49.73%
A-F3 DRY	67.42	13.76%	–1.77%	–2.18%	7.58%	–1.01%	–1.28%	11.77%
B-F2HUM	24.53	0.43%	0.12%	0.77%	27.49%	1.75%	3.14%	39.59%
B-F2 DRY	40.66	10.65%	–2.12%	–2.48%	12.37%	–0.61%	–0.74%	18.37%
B-F3HUM	28.54	0.10%	1.75%	2.66%	57.64%	3.50%	5.68%	66.74%
B-F3 DRY	63.46	26.30%	–9.63%	–9.61%	7.73%	–8.08%	–8.10%	14.50%
A + B-F2HUM	49.12	10.13%	–1.40%	2.08%	57.64%	0.37%	3.58%	68.80%
A + B-F2 DRY	96.09	31.47%	–7.39%	–7.86%	15.61%	–5.81%	–4.68%	20.03%
A + B-F3HUM	57.62	16.54%	–4.84%	–4.37%	98.53%	1.87%	–2.38%	120.59%
A + B-F3DRY	141.18	39.09%	–14.12%	–10.84%	23.84%	–13.08%	–8.84%	28.94%
Average			–3.27%	–2.50%	32.00%	–1.46%	–0.24%	41.72%

*Note: *These solutions are proven optimal by CPLEX.*

In Table 3, the best distance—D according to Eq (1)—obtained for each real instance with CPLEX and each metaheuristic is reported. Besides, it is reported the gap estimated by CPLEX to the ideal solution calculated by the software and the percentage difference between the best solution obtained by each metaheuristic and the solution obtained by CPLEX according to Eq (9).

$$\Delta^{MH} = \frac{D^{MH} - D^{CPLEX}}{D^{CPLEX}}\% \tag{9}$$

where D^{MH} is the distance obtained by each metaheuristic and D^{CPLEX} is the distance obtained with CPLEX. Thus, in Table 3 when Δ^{MH} has a negative value it means that the metaheuristic obtained a better result than CPLEX. We compare the CPLEX solution with two distances of the metaheuristic: the best or minimal distance and the median distance.

In order to evaluate its performance, Table 4 records the number of iterations required by each metaheuristic in the time established by the Eq (8).

Table 4. Average time and iterations required for real-world MSW instances.

Instance	M (sec)	Average number of iterations		
		SA	LNS	GA
A-F2HUM	828	2785036.17	905.97	66449.33
A-F2DRY	828	4234573.77	1623.03	64646.87
A-F3HUM	948	5310001.37	549.90	68082.77
A-F3 DRY	948	5153516.90	828.00	67384.67
B-F2HUM	816	4906570.60	862.00	64281.60
B-F2 DRY	816	2121397.27	1587.80	63497.00
B-F3HUM	1212	7424337.37	371.97	77485.87
B-F3 DRY	1212	7101348.27	591.30	76806.13
A + B-F2HUM	1644	10063074.37	308.03	90283.23
A + B-F2 DRY	1644	9726814.20	121.37	89513.53
A + B-F3HUM	2160	12840985.67	111.60	99323.93
A + B-F3DRY	2160	57603690.73	68.63	98254.67

As aforementioned, to extend the metaheuristic comparison, the instances of the benchmark Set E [40] were solved. Table 5 shows the best results found in each instance by the algorithms under study, while Table 6 shows the performance of the metaheuristics in comparison to the BKS (Best Known Solution) of the instance using the percentage difference estimated with Eq (10).

$$\Delta'^{MH} = \frac{D^{MH} - D^{BKS}}{D^{BKS}}\% \tag{10}$$

where D^{MH} is the distance obtained by each metaheuristic and D^{BKS} is the distance of the BKS of the instance. Again, we compare the BKS with two distances: the best (minimal) value and the median value achieved by each metaheuristic.

Similarly, to the results of real-world instances, in Figure 4 we present a chart with the results of the best distance achieved by each metaheuristic and the BKS of each instance of the benchmark.

Table 5. Results of the benchmark instances of the three metaheuristics.

Instance	SA				LNS				GA			
	min	median	max	iqr	min	median	max	iqr	min	median	max	iqr
E-n13-k4	**247.00**	**247.00**	248.00	0.00	**247.00**	**247.00**	247.00	0.00	247.00	248.00	265.00	1.00
E-n22-k4	**375.00**	**375.00**	383.00	0.00	**375.00**	**375.00**	375.00	0.00	**375.00**	389.00	439.00	14.50
E-n23-k3	**569.00**	**569.00**	597.00	0.00	619.00	619.00	619.00	0.00	**569.00**	628.50	732.00	58.00
E-n30-k3	**503.00**	**503.00**	503.00	0.00	534.00	534.00	537.00	3.00	507.00	547.00	636.00	40.00
E-n31-k7	**379.00**	390.00	425.00	16.00	**379.00**	**379.00**	393.00	0.00	395.00	471.00	573.00	40.75
E-n33-k4	**835.00**	**835.00**	843.00	1.75	**835.00**	**835.00**	835.00	0.00	507.00	547.00	636.00	40.00
E-n51-k5	**521.00**	**521.00**	538.00	5.75	521.00	**521.00**	536.00	0.00	565.00	610.00	681.00	46.75
E-n76-k7	**684.00**	**689.00**	694.00	1.75	830.00	*830.00	853.00	0.00	776.00	*833.50	912.00	59.75
E-n76-k8	**736.00**	738.50	747.00	2.75	737.00	**737.00**	740.00	0.00	846.00	935.00	1061.00	65.75
E-n76-k10	**830.00**	843.00	862.00	7.50	**830.00**	**830.00**	853.00	0.00	930.00	967.00	1052.00	42.25
E-n76-k14	**1023.00**	1034.00	1051.00	9.00	1026.00	**1026.00**	1026.00	0.00	1102.00	1158.50	1651.00	55.00
E-n101-k8	**817.00**	825.50	863.00	5.00	822.00	**822.00**	834.00	0.75	1119.00	1205.00	1365.00	80.00
E-n101-k14	1082.00	1093.00	1127.00	12.75	**1080.00**	**1080.00**	1086.00	0.00	1279.00	1401.50	1570.00	109.50

Table 6. Distances of the solutions of each metaheuristic to the BKS of the benchmark.

Instance	$\boldsymbol{M}$	BKS	Comparison with best $\boldsymbol{D}$ value			Comparison with median $\boldsymbol{D}$ value		
	(sec)		Δ'^{SA}	Δ'^{LNS}	Δ'^{GA}	Δ'^{SA}	Δ'^{LNS}	Δ'^{GA}
E-n13-k4	156	247	0.00%	0.00%	0.00%	0.00%	0.00%	0.40%
E-n22-k4	264	375	0.00%	0.00%	0.00%	0.00%	0.00%	3.73%
E-n23-k3	276	569	0.00%	8.79%	0.00%	0.00%	8.79%	10.46%
E-n30-k3	360	534	*–5.81%	0.00%	*–5.06%	–5.81%	0.00%	2.43%
E-n31-k7	372	379	0.00%	0.00%	4.22%	2.90%	0.00%	24.27%
E-n33-k4	396	835	0.00%	0.00%	1.68%	0.00%	0.00%	9.52%
E-n51-k5	612	521	0.00%	0.00%	8.45%	0.00%	0.00%	17.08%
E-n76-k7	912	682	0.29%	21.70%	13.78%	1.03%	21.70%	22.21%
E-n76-k8	912	735	0.14%	0.27%	15.10%	0.48%	0.27%	27.21%
E-n76-k10	912	830	0.00%	0.00%	12.05%	1.57%	0.00%	16.51%
E-n76-k14	912	1021	0.20%	0.49%	7.93%	1.27%	0.49%	13.47%
E-n101-k8	1212	815	0.25%	0.86%	37.30%	1.29%	0.86%	47.85%
E-n101-k14	1212	1071	1.03%	0.84%	19.42%	2.05%	0.84%	30.86%
Average	–	–	**–0.30%**	**2.53%**	**8.84%**	**0.37%**	**2.53%**	**17.39%**
Average without instance E-n30-k3	–		**0.16%**	**2.75%**	**9.99%**	**0.88%**	**2.75%**	**18.63%**

*Note: *The solutions found have a lower value of D, but a greater number of routes (k = 4) than the reference BKS.*

4. Discussion

In this Section, we discuss the main results from the computational experimentation performed on real waste collection instances and on the benchmark set E. As a summary of the metaheuristics we present Table 7 in which we outlined the main features of the three metaheuristic.

Table 7. Main features of each metaheuristic.

Feature	SA	LNS	GA
Type of algorithm	Local search	Neighborhood search	Evolutionary algorithm
Stopping criterion	Time limit/maximum allowable number of iterations without improvement	Time limit	Time limit/maximum time without improvement
Programming language	VBA	VBA	Python
Main parameters	Initial temperature/ function that determines the cooling mechanism	Auto-tuning by the author	Population size, crossover and mutation probability,

4.1. Real MSW instances

Concerning the real instances of the MSW system of Bahía Blanca, the results are presented in Tables 2–4. Table 2 shows that SA obtained the smallest minimal distance in nine out of 12 instances while LNS provided the smallest distance in the three remaining instances. Regarding the median distances, SA produced the smallest median in eight out of 12 instances (however, in two instances the medians are not found to be statistically different from the ones obtained by LNS). LNS obtained the best median distance in five instances (however, in four instances these medians were not statistically different from those of SA). In instance A-F2DRY both algorithms obtained the same median. Both the SA and the LNS procedures always outperform the GA in terms of minimal values and median values. Neither the GA nor the SA cut were stopped before the maximum allowable computing time due to stagnation.

The results of Table 3 show that CPLEX was able to obtain optimal solutions (with 0.00% CPLEX gap) in only two out of 12 instances, i.e., A-F2HUM and A-F3HUM. In the rest of the instances, the optimality CPLEX gap is larger than zero with the maximum gap in instance A + B-F3DRY (39.09%). Regarding the comparison between the metaheuristics and CPLEX, SA minimal distances are on average 3.27% better than CPLEX solutions, having the largest improvement in instance A + B-F3DRY (–14.12%). Regarding LNS, minimal solutions are on average 2.50% better than CPLEX, having the largest improvement also in instance A+B-F3DRY (–10.84%). The minimal solutions of the GA are on average 32.00% worse than the CPLEX solutions, having the smallest difference on instance A-F3 DRY (7.58%). When comparing the median distances of the metaheuristics with CPLEX solutions, the relationship is maintained, being SA the metaheuristic that obtains the best average difference (–1.46%) followed by the LNS (–0.24%). Results of Table 4 showed that in relation to the average number of iterations carried out, SA and GA reach a greater number of iterations as the number of nodes grows. Conversely, in the case of LNS the number of iterations decreases as number of nodes increases.

Another important feature to analyze is the variation of the results of the algorithms. Since instances were non-parametric, the interquartile range better represents these values. In order to better depict the differences between the algorithms, in Figure 3 we present the box plots of the four real-world representative instances. A box plot is a simple way of visualizing statistical data on a plot in which a rectangle is drawn to represent the second and third quartiles with a vertical line inside to indicate the median value. The lower and upper quartiles are shown as horizontal lines either side of the rectangle. If there are outliers (values that surpass the lower and upper quartiles) these are plotted as extra scatter dots outside of the main box. For the sake of comparison, we also present the unique value of CPLEX. From Figure 3 it can be depicted that the GA always obtains worse results than the other two metaheuristics and CPLEX. Moreover, the variability of the results also seems to be larger, especially in instance A + B-F2HUM (Figure 3(a)). LNS and SA are very competitive against CPLEX. Besides, the variability of these algorithms seems to be relatively small especially in instance A + B-F3DRY (Figure 3(c)).

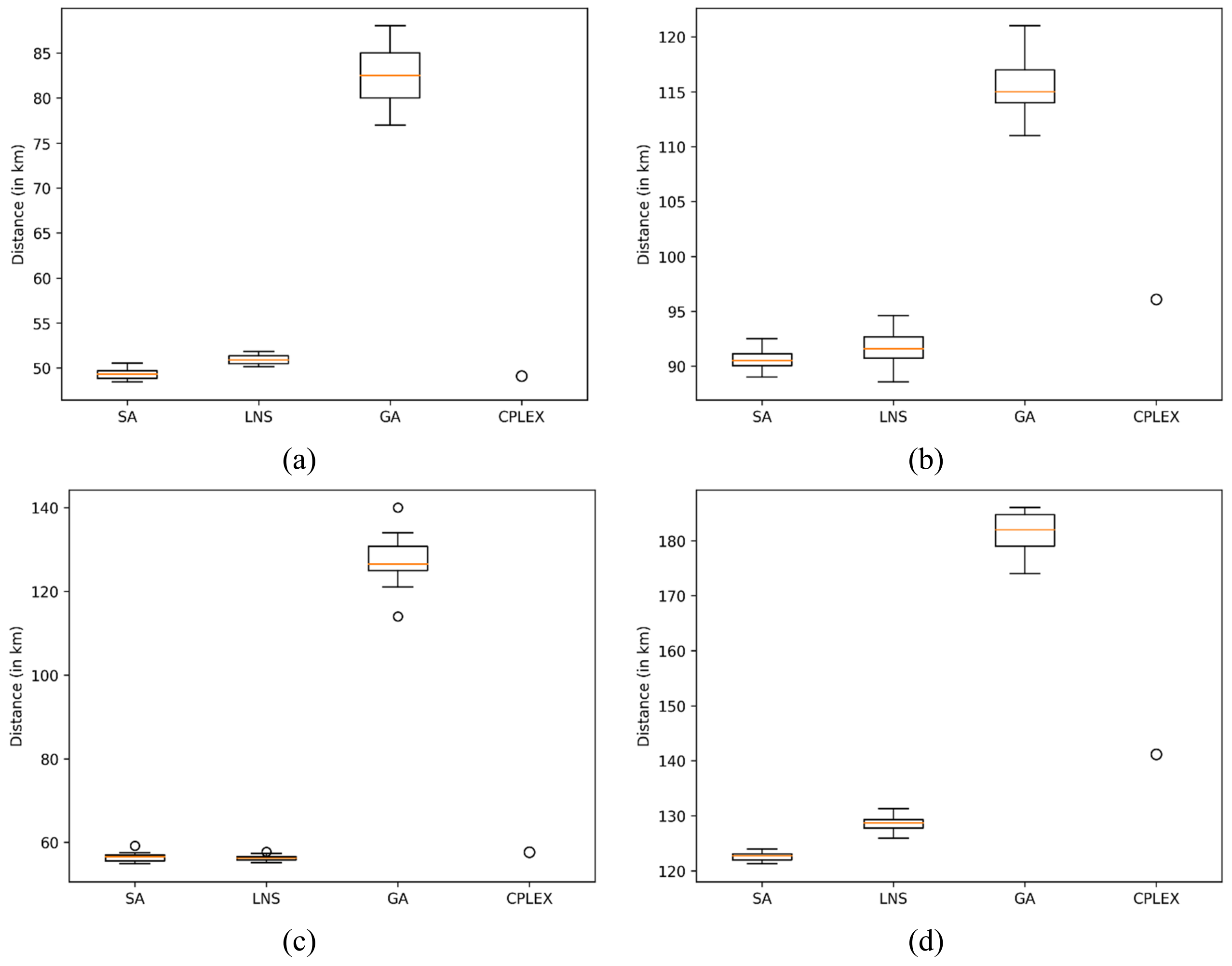


Figure 3. (a). Boxplot of instance A + B-F2HUM. (b). Boxplot of instance A + B-F2DRY. (c). Boxplot of instance A + B-F3HUM. (d). Boxplot of instance A + B-F3DRY. Distance of the best distances in kilometers obtained by each metaheuristic and of CPLEX solution for each real-world instance.

Another aspect that is relevant to analyze in connection with the efficiency of the solution algorithms used to solve the real scenarios, is the impact of the partition of the overall area into two sectors. Partitioning complex problems into smaller (more tractable) parts is a normal procedure employed to address the complexity of NP-hard problems [49], as it is the case of the CVRP that is addressed in this work. In this case, the partition is performed at the level of the input instance dividing the overall area into two smaller Sectors (Sectors A and B). Although this allows the resolution algorithm to face a smaller problem, which might be easier to solve, it can also generate a certain loss of quality of the obtained solution since the algorithm do not take into account the overall scenario.

The joint search space of the partitioned scenarios is smaller than the search space of the overall scenario since some possibilities are not feasible in the partitioned case (i.e., combining bins that belong to different sectors in the same route). For analyzing this behavior for the proposed algorithms, in Table 8 we report the percentage deviation between solving the two partitions separately and solving the global scenario computed according to Eq (11).

$$\frac{D^{A+B} - (D^A + D^B)}{(D^A + D^B)} \tag{11}$$

where D^A, D^B, D^{A+B} are the best distance values for sectors A and B, and the overall area A + B, respectively. Thus, a negative difference determines that the resolution approach is able to improve the results when considering the overall instance in comparison to the partitions.

Table 8. Analysis of performance partition of the real scenario.

Instance	CPLEX	SA	LNS	GA
F2HUM	–1.80%	–4.19%	–0.91%	20.87%
F2 DRY	9.39%	3.56%	2.88%	13.57%
F3HUM	3.35%	–2.70%	–3.38%	39.29%
F3DRY	7.87%	–1.89%	2.08%	24.09%
Average	**4.70%**	**–1.31%**	**0.17%**	**24.45%**

From the results of Table 8, it can be seen that CPLEX is able to improve the solution only in instance of humid waste and collection frequency F2. SA is able to improve the solution when it considers the overall area in comparison to solving the partitions in three out of four cases, having the largest improvement in instances of dry waste and collection frequency F2 (4.19%). LNS improves the solution also in two out of four cases, having the largest improvement in humid waste and collection frequency F3 (3.38%). GA always provides a worse result when dealing with a larger scenario instead of the partitioned problem. Thus, the proposed SA was able to manage quite efficiently the increment in the collection area and take advantage of the enlargement of the search space, providing better solutions for the global scenario. On the other hand, GA could not take advantage of this possibility probably been more affected by the larger complexity of facing a larger scenario.

4.2. Benchmark set E

The results of the benchmark set E are reported in Tables 4 and 5. Results of Table 4 show that, regarding the minimal distances, the SA is able to obtain the best solution in 12 out of 13 instances while LNS in six instances out of 13. The relationship between both metaheuristics is reversed when comparing the median values that in this set of instances are all statistically different according to Kurskall-Wallis test. The medians of the results of LNS are better in 10 instances and those of SA are better in 7 instances. Although it is able to achieve the best minimal value in two instances, GA is usually outperformed by the other two metaheuristics in both minimal and median values.

Regarding Table 5, it is necessary to clarify that both SA and GA find a lower value than the optimal value reported in the literature for the E-n30-k3 instance given that, since they do not have a restriction indicating the number of routes, they report a solution with an additional route. Table 5 shows that the SA is able to obtain solutions that are on average (excluding instance E-n30-k3) only 0.16% worse in comparison to the value of 2.75% of LNS. The efficiency of SA is maintained when considering medians comparison, the solutions being around 0.86% distanced on average from the BKS while the LNS value is about 2.75%. As in the case of the real dataset, the GA provides worse results than those of the other two metaheuristics. Neither the GA nor the SA cut were stopped before the maximum allowable computing time due to stagnation.

Regarding benchmark instances, no boxplots are presented since the results of the metaheuristics have less variability in this set of instances and, thus, the box plots were less illustrative. From Table 6, it can be seen that the interquartile range for most of the results of the LNS and the SA is zero. Only the GA, keeps a certain variability in the results. Instead of box plots, in Figure 4 we present a chart with the results of the best distance achieved by each metaheuristic and the BKS of each instance of the benchmark. Once again, it is seen that the GA usually obtains worse solutions than the other two metaheuristics although there are two exceptions in instances E-n76-k7 and E-n23-k3 (in which GA obtain a better result than LNS). As aforementioned, in E-n30-k3 both SA and GA obtain a better solution but they use a larger number of routes.

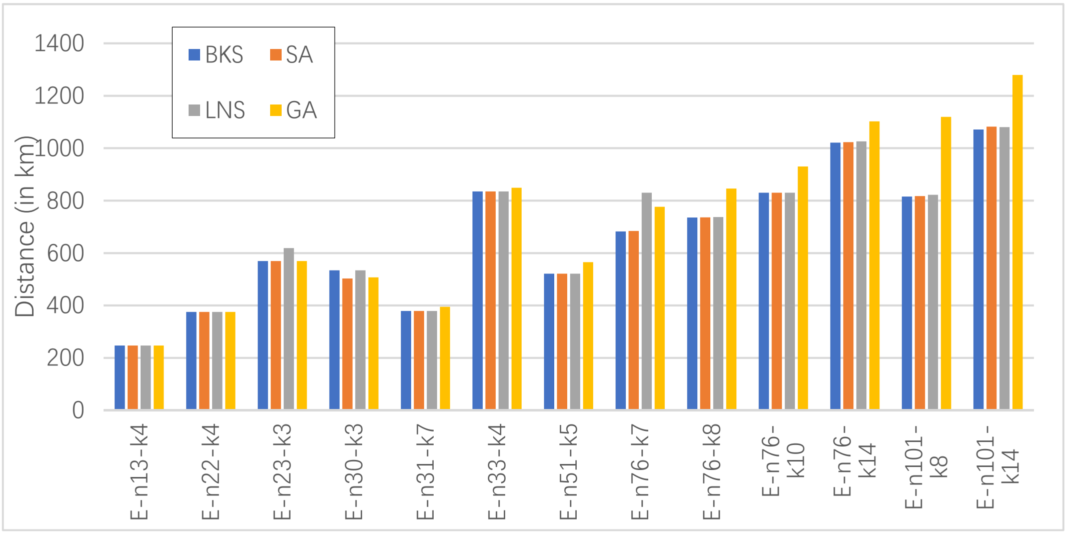


Figure 4. Distance of the best distance obtained by each metaheuristic and of CPLEX solution for each real-world instance.

5. Conclusions

Finding new tools to make municipal/urban solid waste (MSW) logistics more efficient is a pressing concern in today's societies. This work is focused on solving waste collection problems for the Bahía Blanca area. In particular, it addresses a problem found in previous works where the exact solution of waste collection problems was inefficient, because the partitioning of the scenarios into smaller portions was required to achieve convergence to an acceptable feasible solution using reasonable computational times.

Therefore, in this work a comparative study is presented between the exact resolution and three metaheuristic solution tools for the problem of MSW collection in Bahía Blanca. The exact solution is based on a linear programming formulation of the CVRP model using the CPLEX software. On the other hand, with respect to the metaheuristic resolution, a simulated annealing algorithm (SA) is proposed, which is compared with an algorithm taken from the literature based on large neighborhood search (LNS) and a standard genetic algorithm (GA). In addition, the comparison among of metaheuristics using some instances of a well-known benchmark from the literature is carried out. The proposed SA has a similar performance to the LNS algorithm of the literature -obtaining values close to the optimal solution on several occasions- and markedly exceeds the performance of the standard GA.

After analyzing the experimental work carried out, it can be affirmed that the application of these algorithms provides potentially acceptable and competent results, having the metaheuristics the advantage of the less computational effort required to reach the solution. The experience carried out on the benchmark problems and on the MSW instances of Bahía Blanca, marked a good performance of the proposed SA algorithm in comparison to other metaheuristics and CPLEX in both real MSW instances and benchmark instances. It showed to be particularly competitive when compared to a state-of-the-art LNS algorithm from the related literature. On the other hand, GA exhibited a poor performance compared to SA and LNS, possibly due to the use of a standard genetic algorithm with a simple decoding function taken from applications in unconstrained problems such as the Traveling Salesman Problem.

As research lines of work in the future, it is planned to experiment with larger scenarios of the city of Bahía Blanca, using the proposed metaheuristics, which were validated in the present work. It is also intended to incorporate other realistic features to the problem such as time limits for the routes or a finite size of the collection vehicles fleet. Furthermore, it is proposed to develop the coding function of the genetic algorithm with an approach that is better adapted to the type of CVRP problem addressed in this paper. Finally, it is expected to continue performing the comparison between the three metaheuristics with different set of instances and parametric configurations in order to expand the analysis.

Acknowledgments

This work was funded by the research projects PGI 24/J084 and PGI 24/ZJ35 of the Universidad Nacional del Sur. The authors of this work wish to thank V. Herrán Symonds for her assistance in gathering the real data of the MSW scenarios of the city of Bahía Blanca. The third author of this work was funded by the Consejo Interuniversitario Nacional of Argentina (CIN) under the grant Becas de Estímulo a las Vocaciones Científicas (Scholarship EVC-CIN Call 2018).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. D. Hoornweg, P. Bhada-Tata, What a waste: a global review of solid waste management, *World Bank*, **15** (2012), 116.
2. G. Tchobanoglous, F. Kreith, M. Williams, Introduction, in *Handbook of solid waste management* (eds. G. Tchobanoglous, F. Kreith), McGraw-Hill, (2002).

3. E. Tirkolaee, P. Abbasian, M. Soltani, S. Ghaffarian, Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: A case study, *Waste Manage. Res.*, **37** (2019), 4–13.
4. M. Fermani, D. Rossit, A. Toncovich, A simulated annealing algorithm for solving a routing problem in the context of municipal solid waste collection, in *Proceedings of the Xth International Conference of Production Research-Americas 2020*, (2021).
5. P. Pop, L. Fuksz, A. Marc, C. Sabo, A novel two-level optimization approach for clustered vehicle routing problem, *Comput. Ind. Eng.*, **115** (2018), 304–318.
6. C. Miller, A. Tucker, R. Zemlin, Integer programming formulation of traveling salesman problems. *J. ACM*, **7** (1960), 326–329.
7. J. Lenstra, A. Kan, Complexity of vehicle routing and scheduling problems, *Networks*, **11** (1981), 221–227.
8. D. Rossit, J. Toutouh, S. Nesmachnow, Exact and heuristic approaches for multi-objective garbage accumulation points location in real scenarios, *Waste Manage.*, **105** (2020), 467–481.
9. S. Nesmachnow, D. Rossit, J. Toutouh. Comparison of multiobjective evolutionary algorithms for prioritized urban waste collection in Montevideo, Uruguay, *Electron. Notes Discrete Math.*, **69** (2018), 93–100.
10. A. Toncovich, T. Burgos, M. Jalif, Planificación de la logística de recolección de miel en una empresa apícola, in *Proceedings of the X Congreso Argentino de Ingeniería Industrial*, (2017), 397–406.
11. A. Toncovich, D. Rossit, M. Frutos, D. Rossit, Solving a multi-objective manufacturing cell scheduling problem with the consideration of warehouses using a Simulated Annealing based procedure, *Int. J. Ind. Eng. Comput.*, **10** (2019), 1–16.
12. S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680.
13. G. Erdoğan, An open source spreadsheet solver for vehicle routing problems, *Comput. Oper. Res.*, **84** (2017), 62–72.
14. M. Dror, G. Laporte, P. Trudeau, Vehicle routing with split deliveries, *Discrete Appl. Math.*, **50** (1994), 239–254.
15. D. Naddef, G. Rinaldi, Branch-and-cut algorithms for the capacitated VRP, in *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, (2002), 53–84.
16. D. Pisinger, S. Ropke, A general heuristic for vehicle routing problems, *Comput. Oper. Res.*, **34** (2007), 2403–2435.
17. P. Shaw, Using constraint programming and local search methods to solve vehicle routing problems, in *Proceedings of the International Conference on Principles and Practice of Constraint Programming,* (1998), 417–431.
18. C. Groër, B. Golden, E. Wasil, A library of local search heuristics for the vehicle routing problem, *Math. Program. Comput.*, **2** (2010), 79–101.
19. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., (1988).
20. F. Fortin, F. De Rainville, M. Gardner, M. Parizeau, C. Gagné, DEAP: Evolutionary algorithms made easy, *J. Mach. Learn. Res.*, **13** (2012), 2171–2175.
21. M. Haj-Rachid, W. Ramdane-Cherif, P. Chatonnay, C. Bloch, Comparing the performance of genetic operators for the vehicle routing problem, *IFAC Proc. Vol.*, **43** (2010), 313–319.

22. J. Beliën, L. De Boeck, J. Van Ackere, Municipal solid waste collection and management problems: a literature review, *Trans. Sci.*, **48** (2012), 78–102.
23. H. Han, E. Ponce Cueto, Waste collection vehicle routing problem: literature review, *PROMET Traffic Trans.*, **7** (2015) 345–358.
24. B. Kim, S. Kim, S. Sahoo, Waste collection vehicle routing problem with time windows, *Comput. Oper. Res.*, **33** (2006), 3624–3642.
25. A. Benjamin, J. Beasley, Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities, *Comput. Oper. Res.*, **37** (2010), 2270–2280.
26. E. Tirkolaee, I. Mahdavi, M. Esfahani, A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time, *Waste Manage.*, **76** (2018), 138–146.
27. P. Nowakowski, K. Szwarc, U. Boryczka, Vehicle route planning in e-waste mobile collection on demand supported by artificial intelligence algorithms, *Trans. Res. Part D Trans. Environ.*, **63** (2018), 1–22.
28. K. Mekamcha, M. Souier, H. Bessenouci, M. Bennekrouf, Two metaheuristics approaches for solving the traveling salesman problem: an Algerian waste collection case, *Oper. Res.*, **2019** (2019), 1–21.
29. F. Bonomo, G. Durán, F. Larumbe, J. Marenco, Optimización de la Recolección de Residuos en la Zona Sur de la Ciudad de Buenos Aires, *Revista Ingenierıa de Sistemas*, **23** (2009).
30. F. Bonomo, G. Durán, F. Larumbe, J. Marenco, A method for optimizing waste collection using mathematical programming: a Buenos Aires case study, *Waste Manage. Res.*, **30** (2012), 311–324.
31. F. Bertero, *Optimización de recorridos en ciudades. Una aplicación al sistema de recolección de residuos sólidos urbanos en el Municipio de Concordia*, Master thesis, Universidad Nacional de Rosario, 2015.
32. M. Bianchetti, G. Durán, I. Koch, J. Marenco, Algoritmos de zonificación para el problema de la recolección de residuos urbanos: el caso de estudio de una ciudad argentina, *Revista Ingeniería de Sistemas*, **21** (2017), 81–110.
33. G. Braier, G. Durán, J. Marenco, F. Wesner, An integer programming approach to a real-world recyclable waste collection problem in Argentina, *Waste Manage. Res.*, **35** (2017), 525–533.
34. D. Delle Donne, V. Di Tomaso, G. Duran, Optimizing leaf sweeping and collection in the Argentine city of Trenque Lauquen, *Waste Manage. Res.*, **39** (2021), 209–220.
35. A. Cavallin, D. Rossit, V. Herrán, D. Rossit, M. Frutos, Application of a methodology to design a municipal waste pre-collection network in real scenarios, *Waste Manage. Res.*, **38** (2020), 117–129.
36. D. Rossit, S. Nesmachnow, J. Toutouh. A bi-objective integer programming model for locating garbage accumulation points: A case study, *Revista Facultad de Ingeniería*, **93** (2019), 70–81.
37. D. Rossit, F. Tohmé, M. Frutos, D. Broz, An application of the augmented ε-constraint method to design a municipal sorted waste collection system, *Decis. Sci. Lett.*, **6** (2017), 323–336.
38. J. Toutouh, D. Rossit, S. Nesmachnow, Soft computing methods for multiobjective location of garbage accumulation points in smart cities, *Ann. Math. Artificial Intell.*, **88** (2020), 105–131.
39. A. Vázquez, *Ruteo de alta perfomance con OSRM*, 2018. Available from: https://rpubs.com/HAVB/osrm.
40. N. Christofides, S. Eilon, An algorithm for the vehicle-dispatching problem, *J. Oper. Res. Soc.*, **20** (1969), 309–318.

41. J. Mendoza, M. Hoskins, C. Guéret, V. Pillac, D. Vigo, VRP-REP: a vehicle routing community repository, in *Proceedings of Third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization VeRoLog'14*, (2014).
42. IBM, *IBM ILOG CPLEX 20.1 User's Manual*, 2020. Available from: https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-users-manual.
43. GAMS, GAMS Documentation Center, 2021. Available from https://www.gams.com/35/docs/index.html.
44. P. Augerat, *Approche Polyèdrale du Problème de Tournées de Véhicules*, Ph.D thesis, Institut polytechnique de Grenoble, 1995.
45. M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Am. Stat. Assoc.*, **32** (1937), 675–701.
46. O. Cosma, P. Pop, D. Dănciulescu, A novel matheuristic approach for a two-stage transportation problem with fixed costs associated to the routes, *Comput. Oper. Res.*, **118** (2020), 104906.
47. S. Shapiro, M. Wilk, An analysis of variance test for normality, *Biometrika*, **52** (1965), 591–611.
48. W. Kruskal, W. Wallis, Use of ranks in one-criterion variance analysis, *J. Am. Stat. Assoc.*, **47** (1952), 583–621.
49. A. Mahéo, D. Rossit, P. Kilby, A Benders decomposition approach for an integrated bin allocation and vehicle routing problem in municipal waste management, in *Proceedings of the Xth International Conference of Production Research-Americas*, (2020).

Appendix

Results and comparison of statistical tests

In this appendix we present the results of the Shapiro-Wilk normality test and the Kruskal-Wallis statistical test applied to analyze if the medians are equal or not. In the case of the Shapiro-Wilk test the null hypothesis of this test is that the data has a normal distribution and the alternative hypothesis is that the data cannot be assured to have a normal distribution. In the case of the Kruskal-Wallis test the null hypothesis of this test is that the medians are equal and the alternative hypothesis is that the medians are not equal. Both tests were applied with a statistical confidence level of 0.95. Table A.1and A.2 reports from left to right: the instance, the metaheuristic comparison performed, the statistic value of the test, the p-value, the conclusion of the test, and an indication of which median is smaller in case significant differences among medians were found.

From Table A1 it is concluded that in most of the experiments of 30 runs, according to the Shapiro-Wilk normality test, the results have a normal distribution in 43 out of 75 experiments composed by 12 experiments of SA, 12 of LNS and 19 of the GA. Finally, 32 out of 75 cannot be assured to have a normal distribution composed by 13 experiments of SA, 13 of LNS and 6 of the GA.

Table A1. Results of the Shapiro-Wilk test.

Instance	Comparison	Statistic	p-value	Conclusion
E-n13-k4	SA	0.4522	0.0000	Data cannot be assured to have a normal distribution
	LNS	1.0000	1.0000	Data has a normal distribution
	GA	0.4381	0.0000	Data cannot be assured to have a normal distribution
E-n22-k4	SA	0.4983	0.0000	Data cannot be assured to have a normal distribution

Instance	Comparison	Statistic	p-value	Conclusion
	LNS	1.0000	1.0000	Data has a normal distribution
	GA	0.8781	0.0026	Data cannot be assured to have a normal distribution
E-n23-k3	SA	0.4044	0.0000	Data cannot be assured to have a normal distribution
	LNS	1.0000	1.0000	Data has a normal distribution
	GA	0.9604	0.3166	Data has a normal distribution
E-n30-k3	SA	1.0000	1.0000	Data has a normal distribution
	LNS	0.5774	0.0000	Data cannot be assured to have a normal distribution
	GA	0.8827	0.0033	Data cannot be assured to have a normal distribution
E-n31-k7	SA	0.8452	0.0005	Data cannot be assured to have a normal distribution
	LNS	0.3223	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9669	0.4570	Data has a normal distribution
E_n33_k4	SA	0.5701	0.0000	Data cannot be assured to have a normal distribution
	LNS	1.0000	1.0000	Data has a normal distribution
	GA	0.9650	0.4129	Data has a normal distribution
E-n51-k5	SA	0.6876	0.0000	Data cannot be assured to have a normal distribution
	LNS	0.4088	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9517	0.1879	Data has a normal distribution
E-n76-k7	SA	0.9162	0.0215	Data cannot be assured to have a normal distribution
	LNS	0.2898	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9618	0.3441	Data has a normal distribution
E-n76-k8	SA	0.8816	0.0031	Data cannot be assured to have a normal distribution
	LNS	0.3599	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9646	0.4042	Data has a normal distribution
E-n76-k10	SA	0.9708	0.5600	Data has a normal distribution
	LNS	0.2898	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9218	0.0299	Data cannot be assured to have a normal distribution
E-n76-k14	SA	0.9630	0.3686	Data has a normal distribution
	LNS	1.0000	1.0000	Data has a normal distribution
	GA	0.6130	0.0000	Data cannot be assured to have a normal distribution
E-n101-k8	SA	0.6634	0.0000	Data cannot be assured to have a normal distribution
	LNS	0.4862	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9721	0.5970	Data has a normal distribution
E-n101-k14	SA	0.8568	0.0009	Data cannot be assured to have a normal distribution
	LNS	0.4522	0.0000	Data cannot be assured to have a normal distribution
	GA	0.9749	0.6795	Data has a normal distribution
A - F2HUM	SA	0.9707	0.5579	Data has a normal distribution
	LNS	0.9672	0.4648	Data has a normal distribution
	GA	0.9550	0.2297	Data has a normal distribution
A - F2DRY	SA	0.9351	0.0672	Data has a normal distribution
	LNS	0.8530	0.0007	Data cannot be assured to have a normal distribution
	GA	0.9381	0.0807	Data has a normal distribution
B - F2HUM	SA	0.8772	0.0024	Data cannot be assured to have a normal distribution
	LNS	0.7387	0.0000	Data cannot be assured to have a normal distribution

Instance	Comparison	Statistic	p-value	Conclusion
	GA	0.8970	0.0071	Data cannot be assured to have a normal distribution
B - F2DRY	SA	0.7840	0.0000	Data cannot be assured to have a normal distribution
	LNS	0.9556	0.2385	Data has a normal distribution
	GA	0.9543	0.2202	Data has a normal distribution
A+B - F2HUM	SA	0.9576	0.2686	Data has a normal distribution
	LNS	0.9272	0.0414	Data cannot be assured to have a normal distribution
	GA	0.9577	0.2699	Data has a normal distribution
A+B - F2DRY	SA	0.9687	0.5037	Data has a normal distribution
	LNS	0.9788	0.7941	Data has a normal distribution
	GA	0.9559	0.2420	Data has a normal distribution
A - F3HUM	SA	0.9380	0.0804	Data has a normal distribution
	LNS	0.9097	0.0146	Data cannot be assured to have a normal distribution
	GA	0.9516	0.1860	Data has a normal distribution
A - F3DRY	SA	0.9684	0.4956	Data has a normal distribution
	LNS	0.9830	0.8988	Data has a normal distribution
	GA	0.9316	0.0543	Data has a normal distribution
B - F3HUM	SA	0.9150	0.0199	Data cannot be assured to have a normal distribution
	LNS	0.9048	0.0110	Data cannot be assured to have a normal distribution
	GA	0.9361	0.0712	Data has a normal distribution
B - F3DRY	SA	0.9683	0.4936	Data has a normal distribution
	LNS	0.9342	0.0634	Data has a normal distribution
	GA	0.9634	0.3777	Data has a normal distribution
A+B - F3HUM	SA	0.9337	0.0616	Data has a normal distribution
	LNS	0.9588	0.2887	Data has a normal distribution
	GA	0.9448	0.1225	Data has a normal distribution
A+B - F3DRY	SA	0.9576	0.2690	Data has a normal distribution
	LNS	0.9880	0.9769	Data has a normal distribution
	GA	0.9312	0.0527	Data has a normal distribution

From Table A2, it is concluded that in most of the comparisons between experiments of 30 runs, according to the Kruskal-Wallis test, in 68 out of 75 comparisons medians are not equal. Particularly, in the 25 comparisons between LNS vs. SA, in 14 the median of LNS is smaller than of SA, in 7 the median of SA is smaller than of LNS and in 4 no significant difference was found. In the 25 comparisons between LNS vs. GA, in 21 the median of LNS is smaller than of GA, in 1 the median of GA is smaller than of LNS and in 3 no significant difference was found. In the 25 comparisons between SA vs. GA, in 24 the median of SA is smaller than of GA and in 1 the median of GA is smaller than of LNS.

Table A2. Results of the Kruskal-Wallis test.

Instance	Comparison	Statistic	p-value	Conclusion	Result
E-n13-k4	SA vs LNS	5.3636	0.0206	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	15.9076	0.0001	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	28.7221	0.0000	Medians are not equal	Median of LNS is smaller than GA

Instance	Comparison	Statistic	p-value	Conclusion	Result
E-n22-k4	SA vs LNS	6.5455	0.0105	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	36.6350	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	48.0029	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n23-k3	SA vs LNS	55.7773	0.0000	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	43.0556	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	2.0233	0.1549	Medians are equal	—
E-n30-k3	SA vs LNS	53.3937	0.0000	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	50.5859	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	2.8237	0.0929	Medians are equal	—
E-n31-k7	SA vs LNS	24.0734	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	42.4360	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	48.1851	0.0000	Medians are not equal	Median of LNS is smaller than GA
E_n33_k4	SA vs LNS	10.3367	0.0013	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	46.2732	0.0000	Medians are not equal	Median of GA is smaller than SA
	LNS vs GA	50.5859	0.0000	Medians are not equal	Median of GA is smaller than LNS
E-n51-k5	SA vs LNS	4.7031	0.0301	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	45.5182	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	48.1793	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n76-k7	SA vs LNS	49.0410	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.5507	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	0.2650	0.6067	Medians are equal	—
E-n76-k8	SA vs LNS	20.5138	0.0000	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	44.4562	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	48.7022	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n76-k10	SA vs LNS	39.0546	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.3029	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	48.7067	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n76-k14	SA vs LNS	27.2187	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.2906	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	50.5827	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n101-k8	SA vs LNS	9.3821	0.0022	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.3288	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	46.5721	0.0000	Medians are not equal	Median of LNS is smaller than GA
E-n101-k14	SA vs LNS	42.1153	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.2783	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	47.7389	0.0000	Medians are not equal	Median of LNS is smaller than GA
A - F2HUM	SA vs LNS	0.5917	0.4418	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	44.4525	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.4835	0.0000	Medians are not equal	Median of LNS is smaller than GA
A - F2DRY	SA vs LNS	2.4603	0.1168	Medians are equal	—
	SA vs GA	44.6181	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.6907	0.0000	Medians are not equal	Median of LNS is smaller than GA
B - F2HUM	SA vs LNS	23.6180	0.0000	Medians are not equal	Median of SA is smaller than LNS

Instance	Comparison	Statistic	p-value	Conclusion	Result
	SA vs GA	44.4599	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.6469	0.0000	Medians are not equal	Median of LNS is smaller than GA
B - F2DRY	SA vs LNS	0.5579	0.4551	Medians are equal	—
	SA vs GA	44.4724	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.4935	0.0000	Medians are not equal	Median of LNS is smaller than GA
A+B - F2HUM	SA vs LNS	42.5142	0.0000	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.3547	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.3597	0.0000	Medians are not equal	Median of LNS is smaller than GA
A+B - F2DRY	SA vs LNS	10.3878	0.0013	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.3807	0.0000	Medians are not equal	Median of LNS is smaller than GA
	LNS vs GA	44.3807	0.0000	Medians are not equal	Median of SA is smaller than GA
A - F3HUM	SA vs LNS	25.9155	0.0000	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	44.4922	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.5732	0.0000	Medians are not equal	Median of LNS is smaller than GA
A - F3DRY	SA vs LNS	5.3204	0.0211	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.5208	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.5271	0.0000	Medians are not equal	Median of LNS is smaller than GA
B - F3HUM	SA vs LNS	15.1231	0.0001	Medians are not equal	Median of LNS is smaller than SA
	SA vs GA	44.5109	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.5196	0.0000	Medians are not equal	Median of LNS is smaller than GA
B - F3DRY	SA vs LNS	0.0919	0.7618	Medians are equal	—
	SA vs GA	44.3807	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.3906	0.0000	Medians are not equal	Median of LNS is smaller than GA
A+B - F3HUM	SA vs LNS	0.6141	0.4332	Medians are equal	—
	SA vs GA	44.3251	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.3337	0.0000	Medians are not equal	Median of LNS is smaller than GA
A+B - F3DRY	SA vs LNS	44.2635	0.0000	Medians are not equal	Median of SA is smaller than LNS
	SA vs GA	44.3547	0.0000	Medians are not equal	Median of SA is smaller than GA
	LNS vs GA	44.3560	0.0000	Medians are not equal	Median of LNS is smaller than GA