Instituto Superior Técnico

Department of Engineering and Management

# Development of a metaheuristic for the Inventory Routing Problem in waste collection context

João Miguel Silva Moreno

**A B S T R A C T**

Waste collection companies face efficiency problems in their operations, namely the excess of kilometers travelled to empty containers that have low filling levels.

Ramos et al. (2018) introduced the concept of Smart Waste Collection Routing Problem (SWCRP), which defines optimal dynamic routes through the earlier knowledge of the containers' filling rates, which is known through volumetric sensors. The SWCRP was addressed as a Vehicle Routing Problem with Profits (VRPP) with a 1-day time horizon while Morais et al. (2018) addressed the SWCRP as an Inventory Routing Problem (IRP) with a 10-day time horizon. The comparison between these two academic papers concluded that the IRP approach leads to better results.

The present thesis emerged from this context, and its main goal was to develop a problem-solving approach in addressing the IRP regarding high real instances data. After the description of the problem, the solution method was composed by 2 phases: a heuristic one, comprised by container selection and waste collection days; and a metaheuristic, composed by vehicle waste collection routes. The metaheuristic phase was based on neighborhood search algorithms, where multiple destruction degrees and influence radius were tested. The solution method was tested with small instance data and a real case study. The obtained results were compared to the works of Ramos et al. (2018) and Morais et al. (2018), concluding that although the problem-solving approach developed in this work presented solutions which are near optimal (14%), the results were obtained in a substantially lower computational time.

.

**Keywords:** Inventory Routing Problem (IRP), Metaheuristics, Neighborhood Search Algorithms, Collection routes optimization.

## I. Introduction

Waste collection companies face efficiency problems in their operations, namely the excessive distance to visit waste containers with low filling levels. Recent developments in information technologies and communication allow access to real time information about the bins' filling levels through volumetric sensors. However, this information by itself is not enough and it needs to be integrated into the vehicle's routing design. An efficient vehicle routing definition is a competitive advantage for waste collection companies, not only financially (through the reduction of costs or more efficient allocation of resources), but also in terms of competition, through the reduction of the time spent collecting waste. In this context, the current work continues the work done by Ramos et al. (2018), who introduced the Smart Waste Collection Routing Problem (SWCRP), consisting in the definition of dynamic waste collection routes that explore the use of real-time information regarding the bin's fill-level over a predefined time horizon. In Ramos et al. (2018), the SWCRP was considered as a Vehicle Routing Problem with Profits (VRPP), defining the optimal routes for a time horizon of one day. However, as this approach is focused in short term solutions, it may fail to consider the possibility that the profit associated to each route could be higher, if the waste collection occurred later. Therefore, the subsequent work done by Morais et al. (2018) addressed the SWCRP as an IRP with an extended time horizon, allowing profit maximization for the given period. This work aimed to develop a solution method approach to solve the SWCRP as an IRP using metaheuristics and implemented in programming language Python. Section II will present a literature review; Section III the model and solution method approach formulation; Section IV the application of the model in a real case study, and final conclusions in Section V.

## II. Literature Review

### 1) Inventory Routing Problem

An Inventory Routing Problem (IRP) can be described as a combination of inventory management and vehicle routing problems, in which the supplier must decide when to deliver goods to a set of clients which are geographically dispersed, the total amount to deliver and the clients' routing sequence. The main goal is to find a distribution policy that minimizes the total costs composed by transportation, inventory and stock failure costs and guaranteeing that capacity restrictions are met (Elbek and Wøhlk, 2016). The integration of these two problems and subsequent solving offers integrated logistics solutions through the simultaneous optimization of inventory management, vehicle routing and delivery times (Coelho et al. 2014). This combination allows suppliers to reduce production and distribution costs through the coordination of multiple deliveries, and clients do not need to be concerned with inventory management issues, since the supplier is accountable for it.

Coelho et al. (2014) stated that IRP can be classified according to the following criteria: time horizon, structure, route, inventory policy, inventory decisions, fleet composition and fleet size. Time horizon is the period considered when analyzing an IRP model, that can be finite or infinite. The number of customers and suppliers can vary: one-to-one, if there is one supplier and one client; one-to-many, for one supplier and many clients and many to many, with numerous suppliers and clients. Routing can be direct if there is only one customer per route, multiple if there are several customers in the same route and continuous when there are no central depots, such as maritime applications. Inventory policies define preestablished rules to replenish clients, that can be maximum-level policy (where the replenishment level is flexible but limited to the customer's capacity) and order-up-to-level policy, if the replenishment level is limited to the customer's capacity. Inventory decisions, which determine how inventory management is shaped, can have three variations: lost sales, backorder and nonnegative. Nonnegative inventory management decisions are applied in deterministic problems; if the inventory can become negative, then back-ordering occurs, and the subsequent demand will be supplied later in time. If there are no backorders, then the extra demand will be considered as lost sales. Finally, fleet can be homogeneous or heterogeneous and the number of vehicles available can be restricted to one, many or unconstrainted (Coelho et al. 2014).

Coelho et al. (2014) also classify IRP regarding demand information. If information is fully available to the client in the beginning of the planning horizon, the IRP is deterministic; if demand respects a probability distribution and it is known, IRP can be namely as stochastic IRP (SIRP). When demand information is not fully known in the beginning of the planning horizon, but it is gradually revealed over time, this type of IRP can be classified as dynamic inventory routing problem (DIRP).

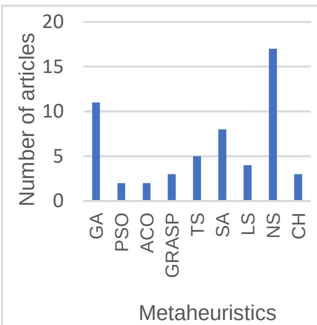
### 2) Metaheuristics

To solve optimization problems, several methods can be used: exact and estimated methods, which include heuristics and metaheuristics. Exact methods constitute exhaustive search algorithms, covering all search space to seek and attain an optimal solution. However, these are not viable when facing problems with larger dimensions because they are also computationally demanding (Archetti et al., 2007).

With the development of complexity theory, the scientific community understood that combinatorial nature problems are NP-hard, therefore, it is pointless to waste resources searching for efficient and exact solutions, which led to the usage of estimated methods, which although guarantee a solution, it is not necessarily the optimal one.

Estimated methods can be divided in two approaches: heuristics and metaheuristics. Heuristics only explore a subpart of the solution space and lead to local optima, avoiding global optimal solutions. Metaheuristics are solution methods linking local improvement procedures and higher level strategies to create procedures that perform a robust search in a solution space and escape from local optima (Glover, 2003).

To understand the trend regarding solution methods to solve IRP, a literature systematic revision was conducted. The revision was split in two phases: first, "30 years of Inventory Routing" by Coelho et al. (2013) was analyzed, and since this article covers all IRP scientific developments between 1983 and 2013, in the second phase the research was focused on the current period (2013-2018). According to Coelho et al. (2013), metaheuristics were gradually adopted rather than exact methods and heuristics. In that period, the metaheuristics most used were Local Search (LS), Variable Neighborhood Search (VNS), Tabu Search (TS), Adaptative Large Neighborhood Search (aLNS) and Greedy Randomized Adaptative Search (GRASP). In the second phase, 65 papers were analyzed to find a metaheuristics trend regarding IRP for the time horizon studied (2013-2018), shown in figure 1.




| Abs. | Description |
|---|---|
| GA | *Genetic Algoritm* |
| PSO | *Particle Swarm Optimization* |
| ACO | *Ant Colony Optimization* |
| GRASP | *Greedy Randomized Adaptive Search Procedure* |
| TS | *Tabu Search* |
| SA | *Simulated Annealing* |
| LS | *Local Search* |
| NS | *Neighborhood Search* |
| CH | *Clustering Heuristic* |

*Figure 1: Metaheuristics trend for solving IRP (2013-2018).*

Through the analysis of figure 1, a growing use of Genetic Algorithm and Neighborhood Search algorithms to solve IRP is observed. For the present thesis a NS variant, Large Neighborhood Search (LNS), was chosen in order to solve collection IRP Valorsul's case study. In the next subsection the principal aspects of LNS will be presented.

### 3) Large Neighborhood Search

LNS algorithms belong to a metaheuristic set that uses the neighborhood concept. This is a kind of iterative metaheuristics, meaning that they start with an initial solution and gradually transform it with searching operators. In this type of metaheuristic, it is necessary to choose the dimension and structure of the initial solution neighborhood, which will determine if the algorithm will focus on a specific area of the solution or if the solution search area will be wider. The broader the solution neighborhood, the better local optima will be found by the algorithm, however it is computationally time consuming, leading to less iterations in the same time period (Ahuja et al., 2002). Large Neighborhood Search (LNS) is a metaheuristic proposed by Shaw (1998) and the neighborhood is defined by a destroy and repair method, as illustrated in figure 2.

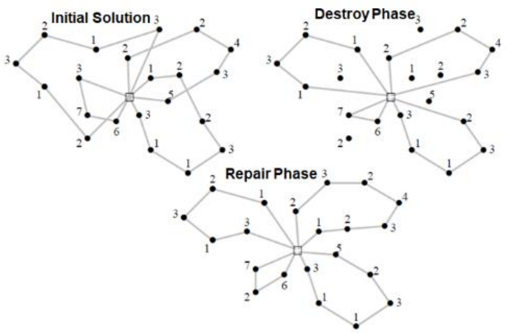



*Figure 2: Destroy and repair method (Pisinger and Ropke, 2010).*

The dots represent several clients and arcs the routes between them.

In the destroy phase, several arcs of the initial solution are destroyed and in the repair phase the same number of arcs are replenished however in a different combination. The pseudocode of the LNS algorithm is presented in the figure 3.

**Algorithm 1** Large neighborhood search

```
1: input: a feasible solution x
2: x^b = x;
3: repeat
4:   x^t = r(d(x));
5:   if accept(x^t, x) then
6:     x = x^t;
7:   end if
8:   if c(x^t) < c(x^b) then
9:     x^b = x^t;
10:  end if
11: until stop criterion is met
12: return x^b
```

| Abs. | Description |
|---|---|
| $x^b$ | Best solution obtained during search |
| $x$ | Current solution |
| $x^t$ | Temporary solution |
| $d(.)$ | Destroy phase |
| $r(.)$ | Repair phase |
| $c(.)$ | Value of objective function |

*Figure 3: LNS pseudocode (Pisinger and Ropke, 2010).*

The temporary solution $x^t$ can be discarded or promoted to current solution if its value of the is higher than that of the current solution (line 6). In line 4, the algorithm applies the destroy phase and then the repair phase, by returning a complete solution $x^t$, with all the arcs positioned in a different sequence comparing to the initial solution. The LNS metaheuristic does not search the whole initial solution neighborhood, it just selects a sample of it. The stopping criteria is defined by the decision maker however temperature (used in Simulated annealing) is the most commonly used (Pisinger and Ropke, 2010).

The destroy phase represents an important part of the LNS implementation. The decision maker has to decide which destruction degree is wanted: if a larger part of the solution is destroyed during this phase, the search space and time consumption will increase but it can led to repeated re-optimization; however if a small part of the solution is destroyed, the solution space will be not explored properly, which can lead to solutions with low values regarding the objective function (such as profit or other performance measure).

In the repair phase, the decision maker can opt by two approaches: the algorithm can lead to an optimal solution (from the initial solution) or a good solution obtained heuristically. An optimal repair operation will be slower than a heuristic one but potentially leading to higher quality solutions in a few iterations. From a diversification point of view, an optimal approach can only lead to superior or identical profit solutions and it can be difficult to leave local optima solutions unless a large part of the initial solution is destroyed (Pisinger and Ropke, 2010).

This work proposes a solution method approach to solve the SWCRP as an IRP by using LNS metaheuristic and will be implemented in programming language Python.

## III. Model Formulation

0This section presents the Valorsul's case study and model formulation. In subsection 1 the case under study and its characteristics are introduced; in subsection 2 the IRP mathematical formulation is presented and finally in subsection 3 the solution method approach developed to address IRP applied to Valorsul's case study is presented.

### 1) Valorsul's case study

Valorsul is a solid waste company that is responsible for the waste management in 19 municipalities in Portugal. This company acts in several phases of waste treatment: collection and sorting of recycling waste, organic valuation and energetic valuation. Valorsul offers integrated solutions for the life cycle of the majority of the types of waste (organic, recycling and undifferentiated). In 2017, Valorsul sent 25 thousand tons of glass, 15 thousand tons of plastic, 30 thousand tons of paper and 921 thousand tons of organic waste, to be recycled. Although Valorsul's area of intervention is less than 4% of the country's size, this company adds value to more than 20% of all domestic waste produced in Portugal (Valorsul, 2018).

Concerning waste collection routes, Valorsul is responsible for 14 of the 19 municipalities it works in, whereas in the other 5, city councils have their own fleets. For that purpose, it provides 8274 containers for the population and has 26 collection routes for paper, 30 for glass and 26 for plastic. Currently, their routes are defined independently of the containers' filling ratio, creating inefficiencies in their operations, since vehicles travel kilometers to collect containers with <25% of their maximum filling ratio.

The Valorsul's case study is composed by two parts: test instances and real instances. The aim of applying the solution method created to test instances is to evaluate if the results are consistent and close to Morais et al. (2018). If so, and if programming errors are inexistent, it is possible to proceed with real instances, in which the solution method is applied to a collection route (68 containers) and to a set of three collection routes (226 containers).

### 2) IRP Mathematical Formulation

IRP mathematical formulation is presented in this section, including notation, sets, parameters and decision variables, preceding the objective function and its constraints.

Sets

$i -$ departure node, $i \in I = \{0,1,2,\dots,n\}$

$j -$ arrival node, $j \in I = \{0,1,2,\dots,n\}$

Node 0 corresponds to the depot, considering that the collection route starts mandatorily on the depot and returns at the end of the route.

t – collection day t of the total set of temporal horizon days, $\in T = \{1,2,\dots,z\}$

Parameters

$C = 1 -$ Travelling cost per distance unit $\left(\frac{€}{km}\right)$ (Source: Valorsul)

$B = 29{,}5 - Waste\ density\ \left(\frac{kg}{m^3}\right)$

$E = 75 - Container's\ capacity (kg) (Source: Valorsul)$

$Q = 4000 - Truck's\ capacity\ (kg)\ (Source: Valorsul)$

$R = 0.0952 -$ Selling price of a recyclable material $\left(\frac{€}{kg}\right)$ (Source: Valorsul)

$d_{ij}$- weighted covered distance between start node I and arrival node j (km)

$A_i - expected\ daily\ accumulation\ rate\ of\ bin\ i\ (kg)$

$S_{i0} -$ Amount of waste (kg) at bin I at the beginning of time horizon, calculated using the information given by the sensor ( $m^3$) and the material density (B)

Decision Variables

$g_{it}$ – Binary variable indicating if waste bin i is visited (1) or not (0), at day t

$x_{ijt}$ – Binary variable indicating if edge ij is visited (1) or not (0), at day t

$f_{ijt}$ – Positive variable representing waste quantity transported by vehicle between container i and j at day t

$w_{it}$ – Positive variable representing the amount of waste collected coming from container i, at day t

$u_{it}$ – Positive variable representing the amount of waste present on the container i, at the end of day t

Objective Function

$$Max\ P = R \times \sum_{i \in I\backslash\{0\}} \sum_{t \in T} w_{it} - C \times \sum_{i \in I} \sum_{j \in I,(j \neq i)} \sum_{t \in T} x_{ijt} \times d_{ij} \quad (1)$$

Constraints

$$\sum_{j \in I,(j\neq i)} f_{ijt} - \sum_{j \in I,(j \neq i)} f_{jit} = w_{it}, \forall i \in I\backslash\{0\}, t \in T \quad (2)$$

$$f_{ijt} \leq Q - A_j \times x_{ijt}, \forall i,j \in I, i \neq j, t \in T \quad (3)$$

$$f_{ijt} \leq (Q - w_{jt}), \forall i \in I, j \in I\backslash\{0\}, i \neq j, t \in T \quad (4)$$

$$f_{ijt} \geq w_{it} - bigM(1 - x_{ijt}), \forall i \in I\backslash\{0\}, j \in I, i \neq j, t \in T \quad (5)$$

$$\sum_{j \in I, i \neq j} x_{jit}, = g_{it}, \forall i \in I\backslash\{0\}, t \in T \quad (6)$$

$$\sum_{j \in I, i \neq j} x_{ijt}, = g_{it}, \forall i \in I\backslash\{0\}, t \in T \quad (7)$$

$$\sum_{i \in I\backslash\{0\}} x_{i0t} = \sum_{i \in I\backslash\{0\}} x_{0it}, \forall t \in T \quad (8)$$

$$w_{it} \leq BigM \times g_{it}, \forall i \in I\backslash\{0\}, t \in T \quad (9)$$

$$u_{it} \leq bigM(1 - g_{it}), \forall i \in I\backslash\{0\}, t \in T \quad (10)$$

$$u_{i0} = S_{i0}, \forall i \in I\backslash\{0\} \quad (11)$$

$$u_{it} = u_{it-1} + A_{it} - w_{it}, \forall i \in I\backslash\{0\}, t \in T \quad (12)$$

$$u_{0t} = \sum_{i \in I\backslash\{0\}} w_{it}, t \in T \quad (13)$$

$$u_{it} \leq E - A_{it+1}, \forall i \in I\backslash\{0\}, t \in T \quad (14)$$

Variable's domain

$$x_{ijt}, g_{it} \in \{0,1\}, \forall i,j \in I, t \in T, i \neq j \quad (15)$$

$$f_{ijt}, u_{it}, w_{it} \in \Re^+, \forall i,j \in I, t \in T, i \neq j \quad (16)$$

The Objective function aims to maximize the profit during the defined time horizon. The profit is determined by the difference between revenue obtained by the selling of collected waste and transportation costs associated to its collection. For all instances, a single vehicle is considered, and all containers have a 2,5 $m^3$ volume. Detailed information about constraints is included in Morais et al. (2018), in which this work's IRP model formulation is based.

### 3) IRP solution method

After presenting the IRP mathematical model applied to waste collection with a time horizon based on Morais et al. (2018), a metaheuristic LNS as a solution method approach was developed. This metaheuristic aimed to, in an iterative approach, find the best collection route sequence for each data instance by respecting the problem's restrictions. The solution method developed is composed by 2 distinct phases: heuristic, to select collection days and containers to collect; and metaheuristic to reach optimization of collection routes, as shown in figure 4.

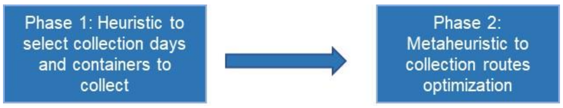



*Figure 4: Main phases of solution method approach developed.*

### 3.1) Phase 1: Heuristic to select collection days and containers to collect

Phase 1 is composed by 2 sub-phases: Selection of collection days (1.1) and selection of containers (1.2). It is on phase 1.1 that the solution method searches for the first day of collection, representing the first day in which at least one bin is in overflow. The heuristic verifies in which day the amount of waste in each container is equal or superior to the container's capacity. If at least one container is in these conditions, the collection route is performed in the day before the overflow day, to guarantee that no bin will be overflowed. If the heuristic does not find any upcoming overflow, a collection route is performed in the last day of the time horizon if the profit is positive. The goal is to verify whether delaying the collection for the last day of the time horizon is profitable. Regarding phase 1.2, 5 alternatives were defined to select the containers to include in a collection route, besides the ones that will be in overflow. Each alternative was developed to verify how much the profit is affected when the containers to collect are chosen using different criteria. In phase 1, the bins' visiting sequence is based on a greedy criterion, in which the next bin to be visited is the one that is geographically closer to the current one. The 1st alternative considers that, after finding an overflow day, all containers in the instance are included on the collection route; the 2nd alternative only includes on the collection route the containers that are overflowing; the 3rd alternative aims to extend the search to containers that are not on overflow but which may bring profit from their collection. This approach is inspired by the work of Mes et al. (2014), which considers the overflowing containers as MUSTGO containers and the other containers in the instance as MAYGO containers. The MAYGO containers can be included into the collection route but must respect 2 criteria defined by the author: influence radius (the distance between each MUSTGO container and each MAYGO container) and positive profit (the profit associated to the selling of waste present on each MAYGO container if the MAYGO containers are converted to MUSTGO). To calculate the profit of each MAYGO container, the distance used is the one between the MAYGO container closest to MUSTGO container. Figure 5 shows MUSTGO containers as green and MAYGO containers as orange. If a MAYGO container is inside a MUSTGO's radium of influence and the profit to visit a MAYGO container is positive, the MAYGO container will be included in the collection route.

*Figure 5: MUSTGO containers (green) and MAYGO containers (orange).*

The 4th alternative includes the containers that are close to overflow and progressively adds the containers with filling ratio between 75% and 100%. The containers with filling ratio between 50% and 75% are included in the collection route up until the vehicle's capacity is completed. Then, the containers with filling ratio between 25% and 50% are included in the collection route up to the vehicle's capacity or when there are no more containers to add. The 5th alternative aims to introduce a randomness degree into the container's selection, meaning that besides the containers that are overflowing, the heuristic adds 50% of the remaining containers in the instance, chosen on a random basis. This process is repeated 10 times and the collection route that has the highest profit is the chosen one. In this phase, the container's sequence routes for all 5 alternatives were defined by using a minimization distance approach: when defining a collection route, the next container to be included in the collection route is the one geographically closer to the current one. For all alternatives, a collection route in the last day of time horizon is tested. When it is profitable, all containers (1st alternative) are collected. After having the collection routes of the 5 alternatives and their performance data (profit, distance, ratio kg/km, collection days), these were stored and will be used on phase 2.

### 3.2) Phase 2: Metaheuristic to collection routes optimization

In the phase 2 the metaheuristic LNS (Large Neighborhood Search) applied to the current thesis was developed. The LNS metaheuristic settles on two distinct phases: the destroy phase, in which part of the current solution (derived from phase 1) is partially destroyed (some arcs that connect the containers are removed) and the repair phase, which connects all containers but with a different visit sequence. According to Pisinger and Ropke (2010), the most important choice when implementing a destroy method is the degree of destruction. If a small part of the solution is destroyed, the metaheuristic will have difficulties in exploring other solution areas. On the contrary, if a big part of the solution is destroyed, the computational time to obtain the solutions will increase. The intention is to find a destroy method that explores the solution area but is not fixated at local optima and with good values of profit and computational time. In the present thesis, the destroy method chosen was a random method, consisting in removing a percentage of containers presented in the initial solution (originating from phase 1) randomly. By using a random deletion, the removals are typically different between iterations, which allows to diversify the solution search space. The basic greedy heuristic was chosen to perform the repair collection routing process in the repair phase. According to Pisinger and Ropke (2007), a basic greedy heuristic inserts the removal container in the route position which guarantees higher profit. In the current work, the basic greedy heuristic rebuilds the destroyed route by testing all removed containers in all possible positions and calculates the profit associated to each route. For example, if there are 3 removed containers, the first removed container is tested in all positions of the destroyed route; the second removed is tested in all positions and then the third. The container that guarantees the highest profit of these 3 tests will be the one who will be inserted. Afterwards, the remaining containers that were not inserted are tested in all positions and the one who guarantees higher profit will be inserted. The process ends when all containers that were removed are inserted on the positions that generate a route with a higher profit than the initial one (built using a distance minimization distance approach).

According to Ropke and Pisinger (2006), a method to avoid that the heuristic gets caught in local optima is to accept solutions with inferior profit than the current one. Thus, a criterium from simulated annealing is used. A parameter called temperature (T) decreases at each iteration by multiplying a cooling rate c that according to Ropke and Pisinger (2006) assumes the value of 0.99975. The likelihood of accepting worse solutions than the current one in terms of profit is defined by equation 17.

$$p = e^{-(\frac{f(s')-f(s)}{T})} \tag{17}$$

When the profit of the current solution is poorer than the one in the initial solution, a random number between 0 and 1 is generated and if p is superior than r, the current solution is accepted.

## IV. Application to Valorsul's case study

The solution method approach developed is then applied to Valorsul's case study. The results are divided in 3 parts: subsection 1 focuses on test instances, subsection 2 with Valorsul's case study (one route – 68 containers) and subsection 3 with Valorsul's case study (three routes – 226 containers). The solution method approach is implemented using Python 3.7 and all the results are obtain using an Intel® Core i7 4720HQ CPU @ 2.60 Hz. To perform a sensibility analysis and compare to the results of Morais et al. (2018) and Ramos et al. (2018), different scenarios are addressed by variating the degree of destruction and influence radius (3rd alternative). In each subsection the results of all scenarios and the comparison with the 2 works mentioned ahead are present.

### 1) Test Instances

In this section is present the results of the solution method approach developed applied to 10 test instances developed by Morais et al. (2018). The goal is to test the solution method and verify if programming errors are present and if the results are close to the ones obtained by Morais et al. (2018). The number of containers vary between 7 and 13 and the time horizon considered for all test instances is 10 days.

For test instances, 6 scenarios regarding sensibility analysis were considered: destruction degree of 50% (scenario 1) and 70% (scenario 2) and influence radius of 0.5 km (A), 1km (B) and 2km (C). 10 iterations of destroy and repair method for each route (associated with each collection day) were performed. In table 1 and 2 the results of all runs regarding all scenarios for test instances and the computational time associated are present, as well as the profit variation between phases and the computational time variation between destruction degrees, for all test instances and scenarios. The alternative 1 is the one that performs the highest profits across almost all instances. That happens because in this alternative all containers are collected when at least one is overflowing, and there are less containers which are geographically close to each other.

*Table 1: Results of the best alternative after phase 2 (profit and computational time).*

| Test instances | Results after phase 2 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1A | Comp. Time (s) | 1B | Comp. Time (s) | 1C | Comp. Time (s) | 2A | Comp. Time (s) | 2B | Comp. Time (s) | 2C | Comp. Time (s) |
| Instance 1 | 11,26 | 5 | 11,26 | 5 | 11,26 | 4 | 11,26 | 5 | 11,26 | 6 | 11,26 | 7 |
| Instance 2 | 15,88 | 13 | 15,14 | 13 | 14,93 | 6 | 15,88 | 20 | 15,12 | 18 | 14,75 | 23 |
| Instance 3 | 41,14 | 28 | 39,4 | 29 | 38,89 | 30 | 41,14 | 43 | 39,4 | 44 | 38,89 | 48 |
| Instance 4 | 59,93 | 17 | 59,93 | 13 | 59,93 | 14 | 59,93 | 24 | 59,93 | 25 | 59,93 | 31 |
| Instance 5 | 50,52 | 19 | 50,56 | 20 | 50,56 | 22 | 50,56 | 26 | 50,6 | 28 | 50,56 | 33 |
| Instance 6 | 32,23 | 10 | 32,07 | 10 | 32,23 | 10 | 32,23 | 14 | 32,23 | 14 | 32,23 | 15 |
| Instance 7 | 4,97 | 7 | 4,97 | 5 | 5,25 | 8 | 5,25 | 10 | 5,25 | 5 | 5,25 | 10 |
| Instance 8 | 54,3 | 17 | 54,88 | 12 | 55,15 | 20 | 54,88 | 24 | 54,88 | 13 | 54,7 | 24 |
| Instance 9 | 45,87 | 15 | 45,87 | 11 | 45,87 | 11 | 45,87 | 21 | 45,87 | 21 | 45,87 | 22 |
| Instance 10 | 21,11 | 11 | 21,11 | 7 | 21,11 | 7 | 21,11 | 13 | 22,21 | 9 | 21,11 | 15 |
| Average | 33,72 | 14,20 | 33,52 | 12,50 | 33,52 | 13,20 | 33,81 | 20,00 | 33,68 | 18,30 | 33,46 | 22,80 |

*Table 2: Profit variation between phases and computational time variation between scenarios with different destruction degree.*

| Test instances | Pofit variation between phases (%) | | | | | | Comp. time variation between destr. degrees (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1A | 1B | 1C | 2A | 2B | 2C | 1A vs 2A | 1B vs 2B | 1C vs 2C |
| **Instance 1** | 12,7% | 12,7% | 12,7% | 12,7% | 12,7% | 12,7% | 0% | 20% | 75% |
| **Instance 2** | 10,6% | 11,1% | 8,7% | 10,6% | 10,3% | 15,0% | 54% | 38% | 283% |
| **Instance 3** | 3,2% | 6,6% | 3,8% | 3,2% | 6,6% | 3,8% | 54% | 52% | 60% |
| **Instance 4** | 1,6% | 1,6% | 1,6% | 1,6% | 1,6% | 1,6% | 41% | 92% | 121% |
| **Instance 5** | 5,3% | 5,4% | 5,4% | 5,4% | 5,5% | 5,4% | 37% | 40% | 50% |
| **Instance 6** | 23,9% | 23,3% | 23,9% | 23,9% | 23,9% | 23,9% | 40% | 40% | 50% |
| **Instance 7** | 0,4% | 0,4% | 6,1% | 6,1% | 6,1% | 6,1% | 43% | 0% | 25% |
| **Instance 8** | 4,7% | 5,5% | 6,3% | 5,5% | 5,5% | 5,4% | 41% | 8% | 20% |
| **Instance 9** | 2,4% | 2,4% | 2,4% | 2,4% | 2,4% | 2,4% | 40% | 91% | 100% |
| **Instance 10** | 26,9% | 26,9% | 26,9% | 26,9% | 31,7% | 26,9% | 18% | 29% | 114% |
| **Average** | 9% | 10% | 10% | 10% | 11% | 10% | 37% | 41% | 90% |

By analyzing table 1 and 2, we can verify that the maximum variation between phases is 30% and the average is 10%, which is a substantial profit increase.

Regarding destruction degrees, when destruction degree increases up to 70% the profit has small variations. The test instances have few containers and the destruction degree doesn't influence the profit, so it is necessary to test the solution method approach with real instances to verify if there is a tendency regarding the influence of destruction degree in the profit. Table 3 presents the comparison between the results obtained using the method solution approach developed and the results of Morais et al. (2018), for test instances.

*Table 3: Comparison between the results obtained using the solution method and Morais et al. (2018) to solving IRP, for test instances.*

| Test instances | Morais *et al.* (2018) | | LNS (Phase 2) | | Variation (%) | |
|---|---|---|---|---|---|---|
| | Profit | Compt. Time (s) | Profit | Compt. Time (s) | Profit | Compt. Time (s) |
| **Instance 1** | 12,2 | 11,5 | 11,26 | 5 | -7,7% | -56,5% |
| **Instance 2** | 17,2 | 38,4 | 15,88 | 13 | -7,7% | -66,1% |
| **Instance 3** | 42,6 | 1454,9 | 41,14 | 28 | -3,4% | -98,1% |
| **Instance 4** | 60,1 | 1696,3 | 59,93 | 17 | -0,3% | -99,0% |
| **Instance 5** | 51,4 | 1929,2 | 50,6 | 28 | -1,6% | -98,5% |
| **Instance 6** | 40,7 | 198 | 32,23 | 10 | -20,8% | -94,9% |
| **Instance 7** | 39,4 | 568,8 | 5,25 | 8 | -86,7% | -98,6% |
| **Instance 8** | 56,1 | 1593,7 | 55,15 | 20 | -1,7% | -98,7% |
| **Instance 9** | 46 | 4503,8 | 45,87 | 15 | -0,3% | -99,7% |
| **Instance 10** | 34,7 | 1853,8 | 22,21 | 9 | -36,0% | -99,5% |
| **Average** | 40,63 | 1384,84 | 35,26 | 16 | -14% | -90% |

Observing table 3, we see that the profit obtained by using the solution method approach is always inferior than the results obtained by Morais et al. (2018) but the variation, in average, is 14%. Since all alternatives considered do not represent all ways of collecting containers, this means that there may exist other more ways to perform the collection which are more efficient. Regarding computational time, the results of the solution method approach show substantial improvements, in average, less 90% computational time than Morais et al. (2018). However, only with real instances will be possible to evaluate the real impact of the solution method approach.

### 2) Valorsul's case study – 68 containers

In this section the results of the solution method approach applied to the Valorsul's case study - one route (68 containers) are presented. The time horizon for this case study is 15 days and for the period between January 9th and 23rd, this route was performed 2 times, with a combined profit of 111,1 €. Valorsul's current operations do not have fill rate sensors so, in order to simulate that data, in the works of Morais et al. (2018) and Ramos et al. (2018) the data provided by Valorsul's collecting team was used, registering, during collection, the containers' filling rate in 5 categories: empty (0%), less than half (25%), half (50%), more than half (75%) and full (100%). To calculate the containers expected daily accumulation rate, Morais et al. (2018) divide the sum of the filling rate of each bin by number of days and multiply by the time interval between collection routes. For 15 days, the Morais et al. (2018) results for solving the IRP problem predicted 4 routes in days 1,7,9,15 with a combined profit of 184,8 € and a computational time of 14747,7 seconds. The Morais et al. (2018) solution method didn't have the capacity to solve the static IRP. To overcome this problem, a rolling horizon approach was addressed when, in each planning day, the static IRP is solved considering a partial planning horizon.

Considering the solution method approach developed, 6 scenarios for a sensitivity analysis were considered, with a destruction degree of 50% (scenarios 3) and 70% (scenarios 4) and a influence radius (3rd alternative) of 1km (A), 2 km (B) and 5 km (C). 5 iterations of destroy/repair method associated with each route for each alternative were performed and the results of the best iteration, for all alternatives, is presented in table 4.

*Table 4: Scenario results of the application of solution method approach to Valorsul's case study (one route).*

| Profit per scenario | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | Number of containers | 3A | 3B | 3C | 4A | 4B | 4C | Average |
| **Valorsul's case study** | 68 | | | | | | | |
| Alternative 1 | | 80,98 | 95,02 | 87,33 | **114,07** | **115,30** | **113,02** | 100,95 |
| Alternative 2 | | -11,50 | -15,37 | -15,24 | -2,06 | -5,93 | -2,55 | -8,78 |
| Alternative 3 | | 36,16 | 57,69 | 79,37 | 40,41 | 61,24 | 83,99 | 59,81 |
| Alternative 4 | | **83,59** | **99,53** | **99,43** | 102,14 | 100,35 | 100,82 | 97,64 |
| Alternative 5 | | -1,86 | 21,27 | -1,35 | 7,95 | 8,76 | 0,08 | 5,81 |
| Computational time (s) | | 3140 | 3594 | 2485 | 3912 | 4080 | 4059 | 3545 |

By analyzing table 4, the alternative 1 presents higher profit amongst all scenarios and it is the best alternative for scenario 4. In this case, it is profitable to collect all containers, even the ones who have less than 25% of filling rate. As expected, the profit improves when the influence radius increases (3rd alternative). An increase of the influence radius to 10 km and 20 km was tested but the profit did not suffer any changes. The radius is enough to geographically cover more containers, but it is not profitable to collect them. In table 5 a comparison between profit and computational time of the best alternative between scenarios with different destruction degrees is presented, and shows that when the destruction degree increases, the profit improves, reaching 36%. The increase of data between test instances and this case study allows to understand the impact of destruction degree in profit, concluding that the profit improves when the destruction degree increases.

*Table 5: Profit and computational time variation between destruction degrees.*

| Profit variation between destruction degrees (%) | | | Computational time variation between destruction degrees (%) | | |
|---|---|---|---|---|---|
| 3A vs 4A | 3B vs 4B | 3C vs 4C | 3A vs 4A | 3B vs 4B | 3C vs 4C |
| 36,5% | 15,8% | 13,7% | 25% | 14% | 63% |

Another way of obtaining the final solution was tested: split the time horizon in 3 parts (5+5+5 days) and allow mixed alternatives in each part. The 2nd and 3rd part of the time horizon use the data of the best alternative of the previous part. For all scenarios, the one who present highest profit is scenario 4C, with a combined profit of 149,42 €. In 1st part, alternative 4 scored the highest profit (111,81 €) with a collection in day 1 (52 containers); in 2nd part, the alternatives that scored the highest profit are alternative 3 in day 7 (6 containers) and alternative 1 in day 10 (68 containers), with a combined profit of 37,61 €. In the 3rd part, no collection routes were performed. In table 6 the values of profit and computational time for the current situation, dynamic IRP and LNS (mixed alternatives) are shown.
Besides that, a profit variation and a computational time variation between LNS and the 2 scenarios are displayed.

*Table 6: Profit of current situation, Morais et.al. (2018) work (scenario 2) and LNS, a comparison in terms of profit and computational time between LNS (mixed alternatives) and current situation and Morais et al. (2018).*

| Valorsul's case study: 68 containers | | | | |
|---|---|---|---|---|
| Scenarios | Profit | Comp. Time (s) | Profit variation with LNS (%) | Comp. Time variation with LNS (%) |
| Current situation | 111,10 | | 25,6% | |
| Dynamic IRP (Scenario 2) | 184,80 | 14747,7 | -19,0% | -72,0% |
| LNS (mixed alternatives) | 149,42 | 4059 | | |

By analyzing table 6, the solution method approach presents a superior profit of 25,6%, when comparing with the current situation. When comparing with Morais et al. (2018) work, the best LNS mixed alternative scores less 19% than the profit of Morais et al. (2018) but is reached in 3,6 times less time (-72%) than Morais et al. (2018) work.

### 3) Valorsul's case study – 226 containers

In this section the results of the solution method approach applied to the Valorsul's case study- three routes (226 containers) selected for analysis developed by Ramos et al. (2018), in which a VRPP model is run for each day of the 30 day time horizon, are presented. As the present work approaches SWCRP as a collection IRP and in order to compare the results between the present work and Ramos et al. (2018), the IRP should consider a 30 days' time horizon. Unfortunately, running an IRP model with 30 days' time horizon is computationally demanding. So, for the comparison between the results obtained using the solution method approach and Ramos et al. (2018), only the results for the first 8 days are considered, which is the time horizon used in this case study. For the 8-day time horizon (3rd – 10th January), Valorsul performs 4 collection routes on days 1,7, and 8 (two different routes) with a combined profit of 234,7 €. Since the results for Valorsul's case study- 68 containers showed that the 70% destruction degree is the alternative that scores highest profits, for the 226 containers case study only 70% destruction degree is considered for the sensibility analysis. Two scenarios are addressed, one with an influence radius of 10 km and other with an influence radius of 20 km, both with a destruction degree of 70%. 5 iterations of destroy/repair method associated with each route for each alternative were performed and the results of the best iteration, for all alternatives, is presented in table 7.

*Table 7: Scenario's results of the application of solution method approach to Valorsul's case study (three collection routes).*

| Profit per Scenario | | | |
|---|---|---|---|
| Instance | Number of containers | 5A | 5B |
| **Valorsul's case study** | 226 | | |
| Alternative 1 | | | |
| Alternative 2 | | -111,26 | -111,26 |
| Alternative 3 | | 115,29 | 114,35 |
| Alternative 4 | | **217,32** | **220,12** |
| Alternative 5 | | -378,67 | -251,95 |

Alternative 1 is unfeasible: collecting all containers when at least one is close to overflow always exceeds the truck capacity, since this real instance has 226 containers. The alternative 2 presents negative values because only collects the containers that are close to overflow which can lead to a collection route with only one container. Regarding alternative 3, an increment of influence radius between scenario 5A and 5B doesn't reflect on profit, since the differences between those two values are on the random destroy and repair method. An influence radius of 30 km was tested but the profit didn't change - it is enough to geographically cover more containers, but it is not profitable to collect them. The alternative 4 is the one with the highest profits amongst two scenarios and the 5th is the one with worst results. The random inclusion of containers with less than 25% of filling rate and possibly geographically distant from the containers that are mandatorily collected is one possibly reason for the profit results. The scenario 5B is the one that score highest profit (alternative 4 - 220,12 €) with collections in day 1 (79 containers) and day 5 (92 containers). In figure 6 and 7 the collection routes for day 1 and 5 (alternative 4) applied to Valorsul's case study – 226 containers are displayed.

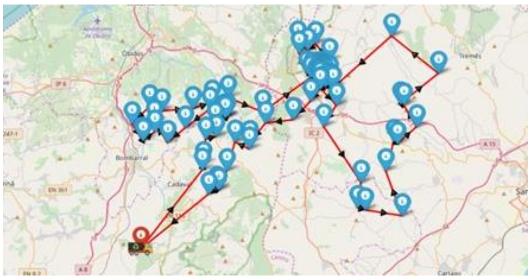


*Figure 6: Collection routes for day 1 for Valorsul's case study - 226 containers.*

*Figure 7: Collection routes for day 5 (alternative 4) for Valorsul's case study - 226 containers.*

For this case study, Ramos et al. (2018) developed two scenarios: scenario 2A (Smart collection approach) and scenario 3A (Smarter collection approach). The scenario 2A selects the containers to collect and its collection sequence guaranteeing the highest profit per each collection day. This scenario doesn't allow overflows and does not consider the following days on time horizon. Scenario 3A combines scenario 2A with a heuristic that defines in which day the solution method should be run to maximize the profit for a given time horizon. This scenario also doesn't allow overflows and does not consider the following days, since it is run every day. These two scenarios were run daily for 30 days but in order to compare with the results obtained using the solution method approach developed, only the first 8 days results were considered. In table 8 the results (profit and computational time) of current situation, scenarios 2A, 3A and solution method approach (LNS) are shown and a comparison between LNS results and the other scenarios are displayed, for Valorsul's case study – 226 containers.

*Table 8: Profit and computational time for current situation, scenario 2A and 3A from Ramos et al. (2018) and LNS, applied to Valorsul's case study (226 containers). Comparison between LNS and current situation, scenarios 2A and 3A.*

| Valorsul's case study - 226 containers | | | | |
|---|---|---|---|---|
| **Scenarios** | **Profit** | **Compt. Time (s)** | **Profit variation with LNS (%)** | **Computational time variation with LNS (%)** |
| Current situation | 234,7 | | -6,6% | |
| Scenario 2A (Ramos *et al*. 2018) | 268,7 | 43200 | -18,1% | -90,3% |
| Scenario 3A (Ramos *et al*. 2018) | 310 | 28800 | -29,0% | -85,4% |
| LNS | 220,12 | 4200 | | |

The profit obtained using LNS is always lower than the analyzed scenarios. Despite that, comparing to current situation, the variation is reduced (6.6%) when comparing with scenarios 2A and 3A, the variation is substantial (18% and 29%). However, computational time for LNS is 4200 seconds (1h10) and for scenarios 2A and 3A is 12h and 8h respectively. Although the profit of LNS is inferior than the scenarios addressed, the solution method approach is computationally more efficient, and justifies the profit difference.

## V. Conclusions

In an era that process improvements are a vital necessity for companies' survival, the companies responsible for waste management face challenges in their collection processes, naming the excess of kilometers cover to collect containers with low filling rate. Thus, access to real time information about containers' filling rate and routes optimization represent an important tool to address the mentioned problem. The present work sought to continue Morais et al. (2018) and Ramos et al. (2018) work developing a solution method approach to solve SWCRP but in order to tackle the problem has used a collection IRP. In order to solve the IRP, the solution method was divided in 2 phases: phase 1 – heuristic for containers selection for collection and collection days and phase 2 – metaheuristic LNS for collection routes optimization. After development and implementation of the solution method approach, it was tested both with test instances and the Valorsul's case study (68 and 226 containers). For the test instances, the destruction degree didn't affect the profit and comparing with the results of Morais et al. (2018), the profits were in average 14%

inferior. For the Valorsul case study – 68 containers, an increment on destruction degree result in an increment of profit in 36% with a computational time increment of 63% (highest value registered). The mixed alternatives presented a profit which was superior than a single alternative along the total time horizon. When comparing with the current situation and Morais et al. (2018) results, the obtained results are 25% superior than current situation and 19% inferior than Morais et al. (2018) but are reached in 72% less computational time, which reflects the main goal of the present work: develop a solution method approach that, for the same data, reaches solutions with similar profits but in an inferior computational time. Concerning Valorsul's case study – 226 containers, the obtained results are inferior than the current situation and Ramos et al. (2018) work. The profit difference between the solution method and the works mentioned increases with the instance data. The difference between the current situation and the obtained results is small (6%) and when compared with Ramos et al. (2018) work, that difference reaches 29% but it is obtained in less computational time (11 hours). A computational time difference of 11 hours justifies the profit variation.

Concerning future works, the suggestion is that the alternatives developed on phase 1 can be revised: although these were developed concerning various possibilities, other scenarios can be addressed to select the containers to be collected. The written code can be optimized, since the author only had a few programming notions before starting the dissertation. The solution method approach was run in a personal computer, so it is suggested to run the method in a workstation, which can reduce further the computational time. It will be interesting to use an ALNS metaheuristic instead an LNS. The ALNS metaheuristic uses diverse destruction and repair methods, which will diverse the solutions in terms of collection sequence, leading to different profit values. Lastly, it is expected that this work shows a new approach to IRP applied to waste collection that, even though it shows inferior profits, solves the problem in less computational time. As stated, there are improvements that can be done (using other metaheuristic or other alternatives' approach) but this work is a solid starting point that can be improved in future works.

## VI. References


Ahuja, R., Ergun, O., Orlin, J. and Punnen, A. (2002) 'A Survey of Very Large Scale Neighborhood Search Techniques', Discrete Applied Mathematics, 123, pp. 75–102.

Archetti, C., Bertazzi, L., Laporte, G. and Speranza, M. G. (2007) 'A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem', Transportation Science, 41(3), pp. 382–391. doi: 10.1287/trsc.1060.0188.

Coelho, L. C., Cordeau, J. and Laporte, G. (2014) '30 years of Inventory Routing', Transportation Science, 48(1), pp. 1–19. doi: 10.1287/trsc.2013.0472.

Elbek, M. and Wøhlk, S. (2016) 'A variable neighborhood search for the multi-period collection of recyclable materials', European Journal of Operational Research, 249(2), pp. 540–550. doi: 10.1016/j.ejor.2015.08.035.

Glover, F. (2003) Handbook of Metaheuristics. 1st Edition. Edited by F. Glover and G. Kochenberger. Springer US. doi: 10.1007/b101874.

Mes, M., Schutten, M. and Rivera, A. P. (2014) 'Inventory routing for dynamic waste collection', Waste Management. Elsevier Ltd, 34(9), pp. 1564–1576. doi: 10.1016/j.wasman.2014.05.011.

Morais, C., Ramos, T. and Barbosa-Póvoa, A. (2018) 'Dynamic approaches to solve the Smart Waste Collection Routing Problem', in Alves, M. J., Almeida, J., Oliveira, J. F., and Pinto, A. A. (eds) Operational Research – Springer Proceedings in Mathematics & Statistics. Springer. doi: 10.1007/978-3-030-10731-4.

Pisinger, D. and Ropke, S. (2007) 'A general heuristic for vehicle routing problems', Computers and Operations Research, 34(8), pp. 2403–2435. doi: 10.1016/j.cor.2005.09.012.

Pisinger, D. and Ropke, S. (2010) 'Large Neighborhood Search', in Handbook of Metaheuristics by M.Gendreau. 2nd Edition. Springer US, pp. 399–420. doi: 10.1007/978-1-4419-1665-5_13.

Ramos, T., Morais, C. and Barbosa-Póvoa, P. (2018) 'The smart waste collection routing problem: Alternative operational management approaches', Expert Systems With Applications, 103, pp. 146–158. doi: https://doi.org/10.1016/j.eswa.2018.03.001.

Ropke, S. and Pisinger, D. (2006) 'An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows', Transportation Science, 40(4), pp. 455–472. doi: 10.1287/trsc.1050.0135.

Shaw, P. (1998) Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, Principles and Practice of Constraint Programming - CP98. CP 1998. Lecture Notes in Computers Science. Edited by G. Goos, J. Hartmanis, and J. Van Leeuwen. Springer. doi: 10.1007/3-540-49481-2_30.

Valorsul (2018) A Valorsul | Valorsul. Available at: http://www.valorsul.pt/pt/seccao/a-valorsul/ (Accessed: 2 July 2018).