NOVA IMS
Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Genetic Algorithm for Waste Collection in Smart Cities

Case of Campolide

Evandro da Silva Mendonça

Project Work presented as requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Title: Genetic Algorithm for Waste Management in Smart Cities

Subtitle: Case of Campolide

Evandro da Silva Mendonça

MGI

2018

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

GENETIC ALGORITHM FOR WASTE COLLECTION IN SMART CITIES

by

Evandro da Silva Mendonça

Project Work presented as requirement for obtaining the Master's degree in Information Management, with a specialization in Knowledge Management and Business Intelligence

Advisor: Professor Miguel de Castro Neto, PhD

October 2018

DEDICATION (OPTIONAL)

ACKNOWLEDGEMENTS (OPTIONAL)

Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org.

ABSTRACT

Abstract text Abstract text.

KEYWORDS

Arc Routing Problem; Genetic Algorithm; Smart Cities; Waste Management

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ARP	Arc routing problem
CARP	Capacitated arc routing problem
GA	Genetic Algorithm
GVR	Genetic Vehicle Representation
ICT	Information and communication technology
OSM	OpenStreetMap
TSP	Traveling Salesman Problem
VRP	Vehicle routing problem

ESSAS SIGLAS SÓ ESTAO NA PRIMEIRA TABELA, DEVEM ESTAR AQUI?

AFR	Africa region
EAP	East Asia and Pacific region
ECA	Europe and Central Asia region
LCR	Latin America and the Caribbean region
MENA	Middle East and North Africa region
OECD	Organisation for Economic Co-operation and Development
SAR	South Asia region

1. INTRODUCTION

1.1. CITIES URBANIZATION AND WASTE MANAGEMENT PROBLEM

Human activity has been pushing environmental changes. Global warming, air pollution, and biodiversity decrease are some of the examples of these changes that can be observed (Bătăgan, 2011). Urban areas are the principal responsible that drive these changes at multiple scales. Being centers of production, consumption and waste disposal, the impacts on the environment can be repeatedly observed among the cities, especially those located in the developed world (Grimm et al., 2008).

The issues generated by the urbanization are even more worrying given that from the 1950s to 2014 the urban population went from 30 percent to more than half of the world's population with 54 percent. Furthermore, in the coming decades, the change on the size and distribution of the urban area will be more expressive, projected to have 66 percent of the entire world's population living in the cities by 2050 (United Nations, 2014). Megacities, the ones that by convention have more than 10 million inhabitants are emerging mostly in the developing world, and economic growth will follow the urban growth, demanding more services and resources (Grimm et al., 2008). Although the urbanization process brings opportunities for development, at the same time challenges arise, namely on social equity, environmental sustainability and government (United Nations, 2014).

One of the major environmental and socio-economic challenges that come with urbanization is waste management (Fujdiak, Masek, Mlynek, Misurec, & Olshannikova, 2016; Karadimas, Papatzelou, & Loumos, 2007). The amount of waste is increasing over time in the urban society. Data from the 2012 World Bank's report shows that the cities were generating about 1.3 billion tons of solid waste per year, costing $205.4 billion. By 2025 it is expected to increase this generation by 2.2 billion tons with the management cost of $375.5 billion, mainly in lower-income countries (Hoornweg & Bhada-Tata, 2012). The projections of urban growth and waste generation for 2025 by region can be seen in table 1.1, the table shows that the regions that currently generates more waste are the one where most of the developed countries are. However, a shifting trend can be observed where the regions with developing countries will take the lead in the waste generation. As an example, the OECD, currently the bigger generator, will increase their waste generation by 11%, and EAP will more than double their waste production.

Waste management is particularly impactful in the short-term to the citizens and the environment, while compared with other problems that massive urbanization may cause (Hoornweg & Bhada-Tata, 2012). The idea of waste management involves many cycles, these can be listed as the collection, transport, processing, recycling, and monitoring. More steps can be presented depending on the cities' waste management scenario, although the waste management aim a common goal in every place it is applied, different cities have their own particularities and need to be addressed in own specific ways. The most important of these cycles naturally is the collection as it directly impacts people living in those urban areas. The collection is also the step that has more costs involved in referring economic terms because it requires intensive labor work and massive use of trucks to be able to deliver the service to the entire city (Beliën, De Boeck, & Van Ackere, 2011). From the total amount of money spent on waste management, 60 to 80 percent is distributed over the collection, transportation, and disposal of solid waste (Karadimas et al., 2007).

VERIFICAR COM O PROF SE A TABELA ESTA BOA E FAZ SENTIDO NA INTRODUCAO

Region	Current Available Data			Projections for 2025			
	Total Urban Population (millions)	Urban Waste Generation		Projected Population		Projected Urban Waste	
		Per Capita (kg/capita/day)	Total (tons/day)	Total Population (millions)	Urban Population (millions)	Per Capita (kg/capita/day)	Total (tons/day)
AFR	260	0.65	169,119	1,152	518	0.85	441,840
EAP	777	0.95	738,958	2,124	1,229	1.5	1,865,379
ECA	227	1.1	254,389	339	239	1.5	354,810
LCR	399	1.1	437,545	681	466	1.6	728,392
MENA	162	1.1	173,545	379	257	1.43	369,320
OECD	729	2.2	1,566,286	1,031	842	2.1	1,742,417
SAR	426	0.45	192,410	1,938	734	0.77	567,545
Total	**2,980**	**1.2**	**3,532,252**	**7,644**	**4,285**	**1.4**	**6,069,703**

Table 1.1 – Waste generation projection for 2025 by region. Adapted from Hoornweg, D., & Bhada-Tata, P. (2012)

Uncollected waste can be harmful to the environment and consequently bring a variety of health issues to the population. Also, poorly waste management have economic impact on the city because the costs can be higher than it would be to properly address the problem. Manage the waste collection of the households is a hard problem that is faced by cities' government across the globe (Hoornweg & Bhada-Tata, 2012).

1.2. SMART CITIES ROLE IN WASTE MANAGEMENT

Cities' main challenge have become be able to manage the ecosystem services dependence, which exhausts the biodiversity and natural resources although prioritizing public health and quality of life (Science for Environment Policy, 2015). With such changes and challenges arising, keeping livable conditions within this context demands a deeper understanding of a smart city, and how it can support cities among the world to deal with these emerging problems (Chourabi et al., 2012).

The smart city concept heavily bases itself on the environmental aspect of the cities and the engagement of people and government in environmental activities (Giffinger, 2007). There is a special motivation on the preservation of natural resources and related infrastructure (Chourabi et al., 2012), and as discussed before, waste management is one of the most important problems with socio-economic impact in the city. Indeed, smart cities itself came to face the challenges that urban areas are facing today and probably the ones that they will face in a near future (Nam & Pardo, 2011).

As others fresh and controversial concepts, the smart city one is not different in the fact that there is no standard definition or template of framing (Nam & Pardo, 2011). In the policy arena in the past years, this concept has been greatly quoted. It seems that the focus approached on this area is about the role of the ICT, an acronym for Information and Communication Technology. The ICT-driven development is believed to be the path to follow for many countries in the EU for example (Caragliu, del Bo, & Nijkamp, 2011).

Obviously, ICT has transformed for better many urban areas economics, social and environment. But laying only in the technology and communication would not benefit the whole city, in some cases, these smart cities need to deal with a problem brought by this form of approaching the concept, like social polarization that create bigger social divisions over the population. The educated and technology included society, mostly middle class, that are attracted by this kind of policy can produce highly gentrified neighborhoods while excluding traditional and poorer residents of the city (Hollands, 2008).

Smarter solutions are arising to deal with the problem of waste management within the smart cities, either using new techniques, data, ICT components, or even a combination of those concepts (Fujdiak et al., 2016). Multiple solutions proposed make heavy use of ICT, as sensors in recycling bins, and brings huge benefits to the waste management saving up time and money spent in the waste collection step (Catania & Ventura, 2014; Fujdiak et al., 2016). Mendes, A. (2017) describe an example of this technology applied in the municipality of Cascais in Portugal. Underground containers are equipped with sensors that indicate data in real time of their level of load. This approach reduced the number of trips that the trucks were used to make to collect the garbage, in consequence, economic saving where attained in the process, the carbon emissions and kilometers traveled also decreased (Mendes, 2017).

The usage of ICT in waste management works well when the garbage is disposed of in fixed bins along the streets, where the truck can make the collect at any time. The door-to-door waste collection has more issues on adopting ICT to improve the collection phrase, an example of a difficulty would be a building with some apartments, in most cases, the residents share the same bins that are collected by a truck in pre-specified days. However, door-to-door type of collection can be addressed in a different way by the smart cities, where the focus is not about using ICT with sensors and technological chips, but using available open data with information of the cities' waste collection generated by the collections routes in the past, and trying to optimize the routes using new techniques to reduce the economic and environmental impact caused by the collection step of the waste management.

1.3. GARBAGE TRUCK ROUTES PLANNING

Intelligent collection management is vital to ensure cost reduction, improve coverage and efficiency of the waste collection process (Buenrostro-Delgado, Ortega-Rodriguez, Clemitshaw, González-Razo, & Hernández-Paniagua, 2015). While focusing on ICT components, one can use many techniques to ensure that the collection is done efficiently, but as stated before, apply this approach when dealing with households' waste collection can be harder than having chips placed on static bins over the streets. Therefore, the collection of waste at each door using ICT is not being considered. Instead, this project's aim is about optimizing the routing of the collection trucks in order to provide a better route for each truck available, while respecting the known limitations.

The collection process is negatively affected by a variety of factors, including poor route planning and the number of vehicles available (Guerrero, Maas, & Hogland, 2013). Optimize the routes that trucks perform in order to serve the households in a city would be beneficial for both the people and the government. Since planning better routes is a well-known hard problem, some algorithms exist in order to give a good enough solution for this issue.

The most common algorithms used in order to deal with the routing problem are the Vehicle Routing Problem (VRP) (Mohammed et al., 2017) and the Capacitated Arc Routing Problem (CARP) (Arakaki & Usberti, 2018), with their variations. The CARP is the key problem inside the Arc Routing Problem (ARP) and is the counterpart to the VRP (Wøhlk, 2008). Both problems are considered hard combinatorial optimization problems (Arakaki & Usberti, 2018; Fadzli, Najwa, & Luis, 2015; HAN & Cueto, 2015; Wøhlk, 2008). These algorithms run upon graphs, that in the waste collection case represents the streets and collection spots. The difference between these algorithms is that the VRP, the most studied routing problem between both (Fadzli et al., 2015; Ramdane-Cherif, 2006), consist in to process the demands of the nodes in a graph, while the CARP focus on serving the edges instead of the nodes (Ramdane-Cherif, 2006). Relating the two algorithms with the waste management problem can be stated that the VRP can be applied to deal with community bins, where each bin is a node in the graph and the edges represent the streets between them. In the CARP, the edges still are the streets and the nodes are intersections between the streets. The CARP approach is more suitable for door-to-door collections since the garbage truck must collect the garbage from a street instead of a specific bin. Indeed it is one of the applications that this algorithm tries to solve (Willemse & Joubert, 2016) because serving the edges instead of the nodes fits better in this problem.

Methods that deal with this kind of collection problem, like vehicle allocation and route designation, can be applied using traditional mathematical methods like linear methods, but these can rapidly suppress the computational resources even with medium sized instances (de Oliveira Simonetto & Borenstein, 2007). This set of problems are known as NP-hard (non-deterministic polynomial-time) problems and until nowadays is only viable to use exact methods for very small instances because of their complexity (Pereira, Tavares, Machado, & Costa, 2002). Therefore, heuristics and meta-heuristics are used to approximate solutions. Although these approaches don't guarantee optimal solutions to the problems, they generally provide good solutions that can be used in real life applications (HAN & Cueto, 2015).

Meta-heuristics has been the path followed by many researchers in the past years, and its result appears to be more promising than heuristics in many cases (HAN & Cueto, 2015). Genetic algorithms have been broadly studied over the literature and seem to allow a more thorough search over the solution space. This approach can lose track of good solutions when jumps to spaces far from optimal solutions, but it also allows moving away from local and not global optimum solutions with some techniques including recombination and random mutations. Some authors are applying genetic algorithms to solve the VRP and ARP with success.

1.4. PROJECT GOALS

With the rise of smarts cities, every aspect of its operation is been rethought. Waste management is one of these aspects that has a huge impact on the city's environment and citizens life. In order to reduce the economic and environmental impact of waste in the cities, better ways to deal with these problems must be found. In fact, many researchers are dealing with these issues, in the academic and corporate world. Waste management by itself is not a sole activity, as stated before, it includes some activities that must be accomplished. Among these activities, this research aims to optimize the routes and usage of trucks in the household waste collection.

The routing problem is a known hard problem and it is contained in the NP-hard problem set. This brings the issue of not been able, until this date, to have an exact algorithm to deal with this problem in a feasible time. Leading the researchers to develop heuristics and meta-heuristics to deal with the problem is a smart way.

The project aims to use GA to optimize the door-to-door collection routes and truck usage with multiple garbage trucks of different capacities, taking into consideration past data from the municipality management to estimate the amount of garbage that each street produces. As stated before, the door-to-door collection problem can be seen as an ARP problem, most specifically the CARP. This algorithm concern is about serving the edges, differently from the usual node serving system. Street information is needed in order to calculate some measures like path length, street direction, and its connections.

To this project, data from the waste collection of several past months is needed. Governments are the largest creator and holder of data within the city (Janssen, Charalabidis, & Zuiderwijk, 2012), and most of the time, they have this data through cities' managers or companies that are responsible for the waste collection. In the case of this project, Câmara Municipal de Lisboa [Posso citar o nome aqui?] shared the data of Campolide's garbage collection, completely without personal data and making sure not to compromise any personal privacy.

OpenStreetMap (OSM) will be used to access the street information. Each piece of route that the garbage truck can travel need to be mapped and stored in a data structure. A usage of this information is to match the data from past collection with each piece of route to estimate the amount of waste generally produced by that place. Also, it allows generating a graph with the connections to measure the distance between two edges.

Having past collection data, and a data structure representing the streets of Campolide will be possible to use a genetic algorithm to calculate multiple routes that the trucks can travel. Choosing the right operators to be applied in the algorithm is a tough step of the genetic algorithm development. Pereira, Tavares, Machado & Costa (2002) created a genetic representation for the VRP algorithm, this representation will be used in this project's genetic operations. With the offspring generation and rules on when to stop the algorithm, the best result of the last offspring will be considered the best solution.

The final solution then will be compared with solutions given by other found in the literature. The result will also be examined against currently existing collection routes of Campolide in order to validate the effort and verify if this project's approach is worthily adopting instead of the existing techniques.

2. LITERATURE REVIEW

2.1. OPEN DATA

Importancia de open data nas cidades a partir do governo e empresas. Falar sobre medidas colaborativas como o OpenStreetMap.

Title	Title
Text	Number
Text	Number
Text	Number

Table 2.1 – Illustrative table

2.2. WASTE MANAGEMENT

Ongoing urbanization stresses the importance of efficiency waste collection, cities must find ways to maximize the acceptance of collection solution (Beliën et al., 2011). Waste collection is about the collection, transportation, and disposal of solid waste from residences, commerce, industry and any other agent that produces solid waste. The collection can be done house-to-house (or door-to-door), via community bins, self-delivered, among others (Hoornweg & Bhada-Tata, 2012). Waste collection is a hard problem that must be aware of many factors that influence the collection, making this step efficient is difficult since this kind of problems does not have an exact solution in a feasible time.

2.3. ROUTING PROBLEM

FALAR SOBRE O CARP PROBLEM

FALAR SOBRE A DEFINICAO MATEMATICA DO CARP

2.4. OVERVIEW OVER GENETIC ALGORITHMS

TENHO QUE MELHORAR ESSA INTRODUCAO

Genetic algorithms, introduced by John Holland (Kumar, Husian, Upreti, & Gupta, 2010) in the early 1970s, belong to the class of evolutionary algorithms and are meta-heuristics based that imitate the biological process of reproduction and natural selection (Carr, 2014). These algorithms are commonly used on search problems and functions optimizer, and it has been applied in a broad range of known problems (Whitley, 1994). One of the greatest barriers of software design, that is to fully understand the structure of complex problems can be solved mimicking natural selection, the specification of every feature of the problems and how to deal with them are not an impediment to search for a solution using this approach (Holland, 1992).

Given its nature, genetic algorithms have been used to find solution for hard problems, like the Travelling Salesman Problem (TSP), VRP, ARP [REFERENCE: ???] and many other problems that due to its complexity don't have an algorithm to give exact solutions. This is possible because these algorithms tends to explore a far greater range of potential solutions in the search space (Holland, 1992).

Because genetic algorithms are based on biological evolution, the terminology used are the same as the one used in biology, although representing fairly simpler concepts than their biological counterpart. Most GA share commons elements, like populations of solutions, selection, crossover and mutation (Mitchell, 1995). To move forward on understanding genetic algorithms, the concepts attached with their nomenclature must be defined, these common elements are described in the list below.

- Gene: a variable (parameter) of the chromosome;
- Chromosome: a set of genes, is a candidate solution for the problem, is the representation of the phenotype on a data structure that can be understood by the algorithm;
- Fitness function: a function to measure the fitness of a solution compared with others, this is the function that must be maximized or minimized depending on the algorithm objective;
- Population: a set of chromosomes that are used to evolve to the next population;
- Crossover: combination of chromosomes to generate offspring for the next generation;
- Mutation: random changes of genes in the chromosome.

Some parameters need to be defined before running the algorithm, they are the population size, mutation and crossover rates.

[REVISAR ESSA PARTE] In principle, a population of individuals selected from the search space, often in a random manner, serves as candidate solutions to optimize the problem [3]. The individuals in this population are evaluated through ("fitness") adaptation function. A selection mechanism is then used to select individuals to be used as parents to those of the next generation. These individuals will then be crossed and mutated to form the new offspring. The next generation is finally formed by an alternative mechanism between parents and their offspring [4]. This process is repeated until a certain satisfaction condition (Jebari & Madiafi, 2013).

2.4.1. Fitness function

Before addressing each step presented at most genetic algorithms, the fitness function must be defined as it is the most important piece used in almost every step of the cycle.

The fitness function is one of the most important part of the genetic algorithm approach as it is the only one method of evaluating the quality of the solution and measure the improvement of through the generations. It must be more sensitive than just measuring good or bad results, it needs to be able to define where the chromosomes stands in the fitness range and compare it with other solutions presented in the population (Carr, 2014).

Fitness function can be the most limitation factor to a genetic algorithm. As addressed above, the fitness function must to translate how to solution performs, this in most cases is not straightforward. Generating complex and expansive fitness functions that are not computational efficient and require hours to complete, as cases of real-world simulations, can be prohibitive in the development of the genetic algorithm. Need to be considered that the fitness function will be evaluated for each chromosome of the population for every new generation produced.

2.4.2. Initialization

The implementation of a genetic algorithm begins with a population with random chromosomes. The size of the population depends on the previous selected size for the population. This size is preserved through the entire life of the algorithm. The initialization can be done totally random or applying some previous knowledge of the problem, in this case, some chromosomes can be included with known genes that makes sense to the problem (Kumar et al., 2010), this can lead the algorithm to converge faster to areas where optimal solutions are more likely to be found. From this early step, the evolutionary process begins.

2.4.3. Selection

A subset of the population is then selected and will be used to breed a new generation, that said, this step is critical since it need to select good individuals trying to keep the diversity of the selected chromosomes. The subset size is also a parameter that need to be set into the algorithm.

The selection step can take place using a variety of techniques. Some methods focus on the fitness of the individual, where chromosomes with best fitness are the one to be selected. Other methods are based on randomness selection or combination of these techniques. No method is guaranteeing to be the best one, and the choice must be problem specific.

There are many selection methods, the most used are roulette wheel and tournament selection (Saini, 2017). But other methods like Stochastic Universal Sampling, Rank Selection and Random Selection can be found in the literature.

The tournament selection and roulette wheel will be addressed bellow. Both methods provide good and diverse parents in most cases, because they give possibility of poorer fit chromosomes to be chosen and still rely on the fitness value to make decision on which individual to choose in their deterministic steps.

2.4.3.1. Tournament Selection

In the tournament selection, the selection process used in this project, K different individuals are randomly selected from the population. Within this set, the chromosome with the best fitness is then selected to reproduce. This process is done once more to select the next parent.

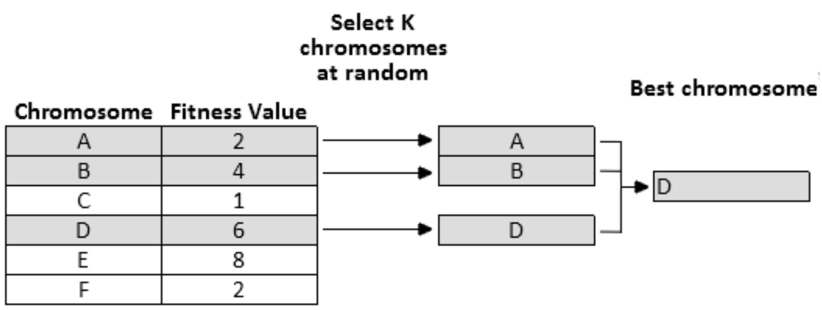

Figure 2.1 – Tournament selection

2.4.3.2. Roulette Wheel Selection

Roulette wheel selection give each chromosome i in the population a probability p(i) of being selected. This probability is proportional with its fitness. To select one individual, a random number is generated, simulating a roulette, and the generated number will define which chromosome will be chose to produce the offspring. Again, this process is done one more time to select the next parent.

2.4.4. Reproduction

Once the parents are selected, the reproduction step takes place. The parents are combined using crossover to generate the offspring. Then, the generated chromosomes can have its genes randomly mutated by the mutation process at a certain rate, this helps the algorithm to run away from local optimum and have a broader exploratory range in the search space. These steps are also problem specifics, giving that each problem will use the crossover and mutation methods that make sense, also their rates.

2.4.4.1. Crossover

Crossover is a vital process in generation new chromosomes. It exchange genetic material (genes) from two or more chromosomes hoping that can generate individuals with better fitness in the next population (A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator). Usually crossover is applied with a high probability, it means that in most cases the genetic material of the parents will be recombined to generate the children, less likely, they will just be copied to the next generation as they are. Using a crossover rate of 100% means that every chromosome in the offspring were generated by crossover at least.

As the selection phrase, there are multiple methods to apply crossover, some of the most known and generic are one point crossover and two point crossover, among others. This project uses a different type of crossover that don't share the behavior of these generic methods and will be further explained in the chapter 4. Because of that, only these two crossover techniques will be explained. These crossover methods will be addressed here to give an overview on how this process take place in the majority of the cases and illustrate the crossover operation.

One point crossover (A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator)

The simplest crossover operator. In this type of crossover, a random point is selected within the limits of the parent, this point is called the cut point. Every point possible to be selected have an equal chance of being selected. To illustrate, in the FIGURE X, two parents chromosomes represented by an array of 10 integers, the cut point would be any number between 0 and 8, in this case, the point 5 was selected. The cut point splits the parents in two half each, the first part are every array element which its index in the array is less or equal the cut point, the second part are the opposite, the elements with index greater than the point. To generate the children, copy the first part of the parent one and insert in the offspring, then get the second part of the other parent and insert in the offspring. Change the order of the parents and do the same operation to generate the second child [REFERENCE: ???].

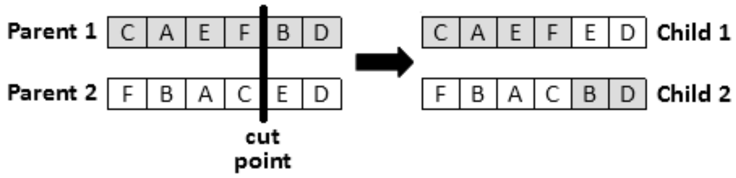

Figure 2.2 – One point crossover

In cases where elements cannot be repeated, the copy of the second parent become a copy of genes one by one in order, avoiding the elements that are already in the child until the child is fulfilled.

Two point crossover

This one is a generalization of one point crossover. The difference between them is that this method chooses two cut point instead of just one, this will split the parents in 3 parts. As a reference, multi point crossover also exists, everything depends on the number of cut points selected.

Two point crossover mix the parts of each parent in the child, the first part of the parent 1 goes first in the child, then the second part of the parent 2 is then inserted, finally, the last part of the parent 1 is inserted. Repeating this operation interchanging the order of the parents generate the second offspring.

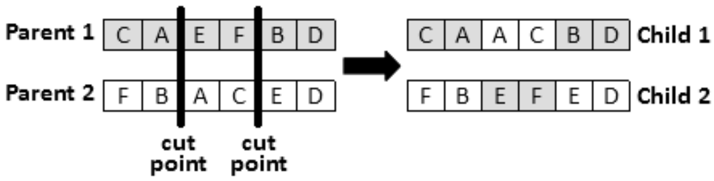

Figure 2.3 – Two point crossover

2.4.4.2. Mutation

Mutation are small random changes in the genetic material of a chromosome. Mutations itself is not supposed to carry the solution to a better fitness in purpose, but they provide an insurance policy against the development of uniform populations that are less likely to improve themselves in the next iteration (Holland, 1992). Typically, the mutation rate is applied with low probability of 1% or less in many cases (Whitley, 1994), because with very right probability, the algorithm could be reduced to a random search over the space.

Common mutation methods are bit flip, swap, inversion, among many other that can be found in the literature. In the example bellow in FIGURE X, the swap mutation is shown. In this mutation technique, two random genes of the chromosome are selected and swapped between them, these cases are useful when repeating a gene is not allowed, as the case of the TSP. In the bit flip mutation, for each gene in the solution, there is a change of change the data in the gene to other random data with possibility to be inserted in the chromosome.

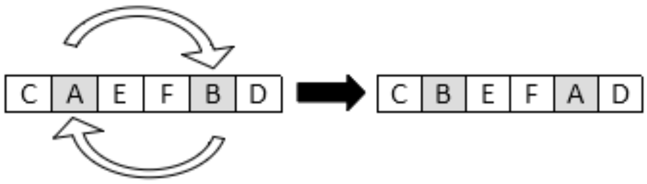

Figure 2.4 – Swap mutation

2.4.5. Termination

Genetic algorithm creates population after population iterating until some condition, or conditions, have been reached. Conditions must be pre-defined by the developer of the algorithm, usually they rely on time, number of iterations, minimum criteria found. Kumar et al. [TA CERTO ESSA CITACAO?] (2010) described some techniques used as stop conditions, they are:

- Found a solution that satisfy a minimum criterion;
- Number of iterations reached;
- Computational time reached (budged);
- The algorithm has reached a highest fitness solution and no longer is producing better solution for an amount of iterations;
- Manual inspection;
- Combination of the previous methods, or any other method created.

The routing problem, both VRP and ARP are hard problems to solve. Among the literature, many researchers try to apply genetic algorithms to find good enough solutions for these problems. Even if the VRP has a wider range of research on it, the methods applied in one can be replicated depending on the representation chosen.

Genetic algorithms for the CARP was created and tested and compared with real case scenarios over the world. To the problem of sprinkler cars routing in Chongqing City in China, taking 37 vertex and 1 deposit into consideration, the GA reduced the travel distance by 33%, also giving better result than other algorithms (Deng et al., 2007).

CARP with time windows GA where developed and compared with heuristics (Ramdane-Cherif, 2006). This research also concluded a superior performance of the evolutionary algorithm when compared with the heuristics.

3. METHODOLOGY [WIP - COLOCAR EM PAST TENSE]

The design science research methodology will be used to accomplish the final goal of this project. By applying this research methodology, the motivation, problem and objectives of the project must be clearly defined in this paper. Then with those steps accomplished, the development of the project will be described, based on the theory previously analyzed. With the project complete, a test case will take place, in the case of this project, an effort will be made using Campolide waste collection data, in Lisbon municipality, Portugal, as a test case of the framework. In the next paragraphs the steps of the methodology applied will be presented with more detail, relating where each piece of the process can be found in this written research.

Following this methodology, in the first chapter are presented the motivation and problem of the study, specifically the subchapters 1.1, 1.2. These subsections give a broadly contextualization in the inherent nature of the population growth problem in the urban areas. Relating it with the sustainability concern in these cities and the waste management problem. The emerging concept of smarts cities to deal with the overpopulation and overgeneration of waste issues are presented, and the collection step of the waste management is approached.

Also, in the chapter one, in the subsection 1.3, the problem statement is presented, on how to optimize waste collection routes on door-to-door collection. This subsection explains the importance and challenges of the routing problem of the waste collection is then described. In the end, genetic algorithms are presented as a possible solution to these kinds of problem.

On the subchapter 1.4 the objectives of this project are defined. On this section, a wide vision of the aim and each step that will lead to the objective is defined, trying to follow a train of thought on how each step connects to each other to accomplish the final aim of having a genetic algorithm to deal with the routing of garbage trucks in a city, using Campolide, at the Lisbon municipality as a model to validate the results.

The problem definition and motivation, besides having some theory to based, was explained in a broader aspect because was not the main proposal of this project. In the chapter 2, the theory used to accomplish the research project will be deeper analyzed having a wider approach of the topics and with more details. This chapter will carry the base theory for the construction of the proposed project. First, on the section 2.1, the concept of waste management will be addressed to have a better understand of what truly waste management is, and how to granulate this concept to the specific target of this project that is the routing of the garbage trucks in the waste collection. In the subchapter 2.2 a literature review will take place on the routing problem using meta-heuristics, focusing on genetic algorithms. A variety of works about the use of genetic algorithms to accomplish the VRP or ARP will be presented. Then, in the subchapter 2.3, genetic algorithm is described, with explanations of its core concepts like generating offspring with the current population and the mutation of existing chromosomes.

[MUST BE BETTER DEFINED, DEPENDS ON THE DEVELOPMENT PROCESS] The chapters *3* to *7* demonstrate the steps of the framework construction, each chapter will approach a specific part of the process. The chapter 3 will discuss about the indicators used in the constructed index.

[TEST CASES MUST BE PLACED HERE]

4. DEVELOPMENT

This project aims to develop a genetic algorithm to deal with the problem of planning efficient routes to garbage trucks in a city. This problem, as stated before, is closely related with the CARP problem. But in real case scenarios, and in the Campolide's case, those trucks don't have all the same capacity CARP-MF. To deal with this difference some tweaks in the algorithms found in the literature must be made. This section gives an overview of the entire development of the project, problems faced, the solution applied to solve these problems and why these solutions were chosen.

The next subchapter will briefly contextualize the tools that were used to assist the problem. This is important to understand the structure of data that are gathered and explain the choices on how to deal with this data.

4.1. A BRIEF WORD ON OPENSTREETMAP

OpenStreetMap (OSM) is a collaborative mapping project where volunteers are free to create and edit geographic map information over the world to an open database. This database is also available for free under the Open Database License.

The street data from OSM, that are relevant to this project, are organized in nodes and ways. The node is the smallest point of data in the map, it represents a single point by defining its latitude and longitude. The nodes can also contain more information inside it, called tags, the tags are used to better qualify the node when it makes sense, for example if the node represent a pedestrian crossing or bus stops, this information will be refereed in tags. Some nodes have no tags, which is not a problem as they are used to represent a path.

```
<node id="21433116" visible="true" version="10" changeset="31019058" timestamp="2015-05-11T21:05:16Z" user="ddtuga" uid="1858517" lat="38.7244842" lon="-9.1772710"/>
```

Node example, user data omitted

Streets in the OSM are represented using one or more ways. Ways are ordered list of nodes, the way direction is defined by the order of the nodes in the way, having that the way starts at the first node and end at the last. Tags are also presented in ways to define things like the street name, the possible directions of the road, among other relevant information. Nodes shared between two or more ways define intersections between streets. Not all nodes are shared, there are nodes that are solely included in one way, for example, a node representing a pedestrian crossing may not also represent a street connection, among other cases.

```
<way id="233939235" visible="true" version="1" changeset="17392454" timestamp="2013-08-18T08:21:23Z" user="Bernhard W" uid="110838">

<nd ref="2422521543"/>

<nd ref="2422521485"/>

<nd ref="2422521542"/>

<tag k="highway" v="residential"/>

<tag k="lanes" v="1"/>
```

```xml
<tag k="name" v="Rua de Campolide"/>

<tag k="oneway" v="yes"/>

</way>
```

Way example, user data omitted

OSM is a powerful collaborative project that allow a variety of applications and studies on its data. There are much more about nodes, ways and how they relate to each other than discussed above, but that are not relevant to this project and therefore will not be addressed here.

In order to develop this project, data from Campolide streets are required, such as streets directions, length and connections, this information can be obtained using OSM. Data from OSM can be exported directly from their main website, but to this project a package called OSMnx was used. OSMnx is a Python package created by Boeing, G. (2017) that facilitates the download of administrative boundaries streets and perform a bunch of useful calculations using data from OSM.

Using OSMnx to download the data, a graph structure is retrieved with the nodes and ways representing the nodes and edges of a graph respectively. OSMnx perform data cleansing process automatically on downloads, the nodes that are not used in intersection and are presented in OSM ways are removed by an algorithm. This result in a simplified graph with just the relevant edges and nodes of a certain location.

Figure 4.1 – Campolide representation with OSMnx

After the download of data from Campolide and Lisbon, both can be stored in the local machine using the library function *save_graphml* to reduce the amount of time required to the upcoming runs of the algorithm, the graph is saved as a GraphML file into the disk.

4.2. Distance Matrix

The method to calculate the distances between edges in this project's GA will follow the distance matrix proposed by Arakaki & Usberti (2018). The first step is to build a distance matrix between edges of the generated graph with Campolide's ways. To build this matrix, each edge from the graph must have its shortest path value to the other edges, in the case of the edge to itself, the distance is defined as the length of the edge.

FIGURA DO GRAFO GERADO

In order to accomplish the creation of this Matrix, Lisbon graph data is needed above the Campolide's one. In a city, from any point that a person is, he or she need to be able to get to any other point [REFERENCE: ???], if this is not possible, would have streets in the city that could not be reached by any means. In this case, Campolide is not an entire city, and is contained inside Lisbon, so it does not provide a path from one edge to all other edges. For that reason, a larger graph that contains the studied graph need to be used to be able to provide valid paths through its edges.

To calculate the distances, the method *shortest_path_length* from *network* package is used. Internally this method uses the Dijkstra's algorithm to calculate the shortest path. The parameters required by the method are the graph, both nodes and the weight that must be taken in consideration, that in this case is the length of the way. The graph provided must be the wider one, the Lisbon graph, because of reasons already approached.

MORE TECHINICAL STUFF HERE NEEDED LIKE: The first step of HGA is to initialize the matrix of distances between required arcs. Let ER be the set of required edges and AR be the set of required arcs, where for each required edge {i, j} ∈ ER there are two corresponding required arcs (i, j), (j, i) ∈ AR. A matrix SP of dimensions |AR| × |AR| is computed such that each entry SP[e, f] is the shortest path cost from the ending node of arc e ∈ AR to the starting node of arc f ∈ AR. For sparse graphs (where |E| is much less than |V| 2) the SP can be computed within O(|V| 3) time and O(|V| 2) space by using the Floyd–Warshall algorithm (Cormen et al., 2001). The SP allows HGA to retrieve the distances between required arcs in O(1) time throughout the optimization process. [Hybrid genetic algorithm for the open capacitated arc routing problem]

FIGURA DE CAMPOLIDE COM AS EDGES FILTRADAS

Although using Lisbon graph as a parameter to the shortest path method, the edges used to iterate the process are from Campolide. Also, only edges that have tags with keys "highways" and values "residential" and "secondary" are taken in consideration as they are the ones where garbage trucks do regular collections. After this filter, 473 edges are left to be calculated, with 274 nodes. Every distance is measured in meters and the distances calculated are not distances between edges but nodes, and nodes distances do not make sense in the CARP problem that serves the edges. To solve this problem, the edges distances are calculated according to each of the shown cases below:

- Same edge: If the edges are the same, it means, the from and to edges are the same edge, the distance will be the total length of the edge. Basically, the distance between the first and last node;
- Different edges: For different edges, will be considered the last node from the first edge, and the first node of the last edge. Added to that are the previous calculated distances of the first and second edges alone. This sum gives the total distance between the two edges.

This process is repeated until every edge in the graph has a distance calculate to each other edge presented. It is a time-consuming task but done only once, with the results stored on the disk, since cities do not change the streets and its orientations often.

CODE

After this calculation, the distances between edges can be retrieved in O(1), as it just need to access the exact point in the matrix. These distances will be heavily used during the GA, so having the information with low complexity worth the time spent at first.

4.3. GENETIC ALGORITHM

Towards the implementation of a genetic algorithm, the representation of the problem, how to measure found solutions and the operations needed in order to generate the upcoming populations must be defined. In the literature review of genetics algorithms was seen the most common representations and operators of a genetic algorithm. This section will discuss how there these apply in this project.

A candidate solution in the population need to the comprehensive, it must carry organized data that allows it to be measured. This is important to the next subchapter that will discuss the fitness function created for this project. In the subsequent subchapters the initialization process and genetic operators will be addressed.

4.3.1. Chromosome Representation

This project will make use of a similar approach to the new GVR (Genetic Vehicle Representation) proposed by Pereira, Tavares, Machado & Costa (2002). Besides the fact that their approach deals with the VRP problem, this can receive small changes in order to fit this project's problem.

The representation must contain the number of vehicles used, the served ways by each truck, as the order of the service. In this representation, the individuals are composed by a list of garbage trucks, each one containing an ordered subset of ways, each way listed in the set is a served edge by the corresponding truck. One solution must contain every edge that need to be served in the problem, independent of where which truck will serve it. Each chromosome is a valid solution to the problem.

Truck				
1	ARC1	ARC2	ARC9	
2	ARC5	ARC3		
3	ARC7	ARC11	ARC6	ARC4
2	ARC12	ARC10		
4	ARC8			

Figure 4.2 – Chromosome representation

In the example above, we have at least four trucks available to allocate in the routes. The number of ways that each truck is serving is related to its capacity. For simplification if every way demand is 1, the truck 1 can serve at least 3 ways, while the truck four at least 1. In the case of the capacity of the truck exceed, a new truck is chosen, and the remaining edges are allocated, the new truck must have enough capacity for at least the first edge that were not able to fit the previous truck, this process is repeated until every edge is allocated in a truck.

To represent this in a data structure in the program, a class representing the chromosome holds two arrays, one is the array with the path followed by the trucks, this array just keeps a sequence of ordered ways disregarding garbage trucks and capacities. The second array keep track of the trucks used in the solution, the first and last ways each one serve, their capacity and the load amount. A truck item in the second array always points to two subsequent indexes in the path array, this is how it keep track on the first and last way that it serves. It's important to note that the index range referenced by the truck item is closed in the left side and open in the right [x1, x2).

Path	ARC1	ARC2	ARC9	ARC5	ARC3	ARC7	ARC11	ARC6	ARC4	ARC12	ARC10	ARC8

Used trucks	(1,5), 0, 3, 3	(2,2), 3, 5, 2	(3,4), 5, 9, 4	(2,2), 9, 11, 2	(4,1), 11, 12, 1

Figure 4.3 – Chromosome data structure

(truck_id, truck_capacity), start_index, end_index, load

Figure 4.4 – Truck data explanation

This combination of arrays provides the information where each truck serves one or more ways in the solution. This representation also allows the genetic operator to act simpler in the paths array, just needing the trucks array to be updated accordingly.

4.3.2. Fitness function

The fitness of a chromosome is defined as the length of all the routes each garbage truck must accomplish. The route of each truck is given by the edges served by them, adding the distance from the deposit to the first edge and the last edge to the deposit. The deposit is a variable containing an edge previously defined in the algorithm, this edge represents the deposit where the trucks start and end their routes.

$$f(s) = d(dep, e_0) + d(e_n, dep) + d(e_0, e_1) + \sum_{i=2}^{n} d(e_{i-1}, e_i) - d(e_{i-1}, e_{i-1})$$ (FALTA AINDA DESCONTAR OS DEPOSITOS)

To calculate the distance between edges the distance matrix previously built is used to retrieve data for the distance function. The distance from the two firsts edges are retrieved and added intro the chromosome fitness. Then, for each edge in the solution skipping the first three edges, the distance between the current edge against the previous edge is calculated. After that, the distance of the previous edge alone is retrieved and subtracted from the sum. This subtraction exists to correct double counts of the distances from the first edge, because the distance matrix stores distances from the beginning of the first edge to the end of the last edge. In the end, the distances from the deposit to the first edge and the last edge to the deposit are considered, and as before, the correction need to be made to remove the double count of the edge's distance. The lower the fitness the best is the solution.

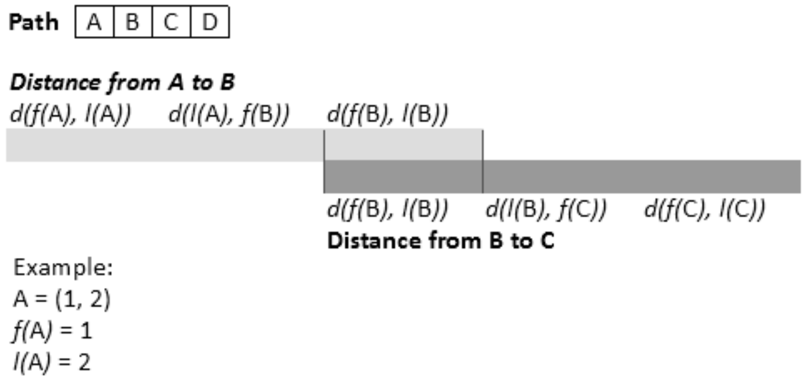

Figure 4.5 – Path distance calculation example

Once the fitness is calculated, it is stored in the chromosome until it suffers some operation that can let the fitness outdated, in this case, the chromosome forgot its fitness and once requested needs to be calculated again. This is done to avoid consecutive calculations of the same routes even if these calculations have a low cost.

4.3.3. Initialization

For the initialization of the population N random chromosomes are generated, N is a variable previously defined in the GA implementation. Each chromosome must represent a feasible solution containing every way that need to be served by the garbage trucks. To generate a random chromosome, every edge that need to be served is randomly shuffled, the result of this shuffle will be the exact order that these edges will be served.

To know which truck will serve the edges, an iteration over the available trucks is done. In each step, a truck in randomly selected from the poll of trucks. With the truck selected, a new ordered iteration over the unserved edges is done, the edges are assigned to the truck until the capacity of the truck is reached, or no more edges need to be served. In the case of the truck wasn't assigned with any edge, it will not be inserted in the solution and no edge will be marked as served, a new truck will be selected, and the process will continue. On the other hand, if the truck actually served one or more edges and its capacity was reached, the truck will be composing the solution with the edges that it serves, and if there are still more edges to be served, the process will continue selecting a new truck.

The iteration method keeps track of the selected garbage trucks and remove it from the selection poll until every truck had the chance to participate in the process, this gives a chance for every truck at least once enter in line to serve an edge (if needed taking into consideration the number or edges to be served). Is not guaranteed by the method that every truck will actually serve an edge, but the aim here is to try to utilize every asset that are available to the task.

4.3.4. Operations

The representation, fitness function and the initialization of the population were defined. To use these definitions to evolve the population to the next generations in a genetic algorithm the operators and how they will act must be defined, the upcoming subchapters will discuss about how the population generate their offspring and evolve over time.

4.3.4.1. Selection

The selection step is when the chromosomes who will act as parents to the new population. These chromosomes will be used for breeding in the crossover step. To select solutions that are diverse among them also relying on how good they are compared to the population, the tournament selection was chosen in this project. The tournament size, that is the number of chromosomes that will participate in the tournament, is a variable previously chosen. The best fit chromosome of the group wins the tournament and is selected to be a parent in the crossover. The process repeats to generate the next parent. This tournament selection does not remove already selected parents from the population, so the same chromosome that just won a tournament can be selected again in the next tournament. In this project the lower distance is the best fitness, them even if very low fitness solution appears in the tournament group, they are very likely to lose against better fit solutions with lower distance, but this also gives a change to not so good solutions, as it can happen to select a group where none of the best fit chromosomes are presented.

Additionally, to always carry over the chromosome from the current generation with the best fitness to the next generation of individuals, a method known elitism or elitist selection will be applied. This method guarantees a spot in the next population to the best individual of the current population, the solution will be carried unchanged, but still can suffer from mutation in the construction of the offspring.

4.3.4.2. Crossover

The crossover operation, as one of the most important operations in a GA, must work with the representation chosen on the chromosomes. In this project acts differently of the common crossover from most of the literature, the genetic material is not exchanged between parents, but a piece of route is given by one of the parents, while the remaining genetic material is totally got from the other parent. This generates only one child in the crossover operations, because of that, more crossovers must be done to complete the next population. This crossover methods follows the one proposed by Pereira, Tavares, Machado & Costa (2002) but applying small changes on how to generate new routes on truck's capacity overflow.

The crossover randomly selects a sub-route of one of the parents and insert it on the other parent. The insertion point is defined to be just after the way which has the minimum distance between itself and the first way in the sub-route. This operation must guarantee that every solution generated from it is a valid solution.

Summarizing the crossover process, two parents previously selected in the selection step of the algorithm and a new chromosome is created to store the data generated by the crossover. A sub-route is randomly selected from one of the trucks of the second parent, this sub-route must have at least one way. The selected sub-route weight is calculated and stored in a variable. One time only, the algorithms verify which edge has the lower distance between itself and the first edge of the sub-route, this verification is done accessing the distance matrix. From now one, the method will go through every truck from the parent 1, and consequently every edge each truck servers. This iteration will append new edges in the child path, and new trucks on the child used trucks array. Its important to notice that if the current edge of the iteration is presented inside the sub-route selected in the first parent, the edge will be not considered, because it would generate duplicated edges when the sub-route is inserted in the new solution. When the iteration finds the edge that is closest to the first edge of the sub-route, the edge will be normally appended in the solution and the sub-route will be inserted just after it. After the insertion of the sub-route in the child chromosome the process continue until every edge is served by the child.

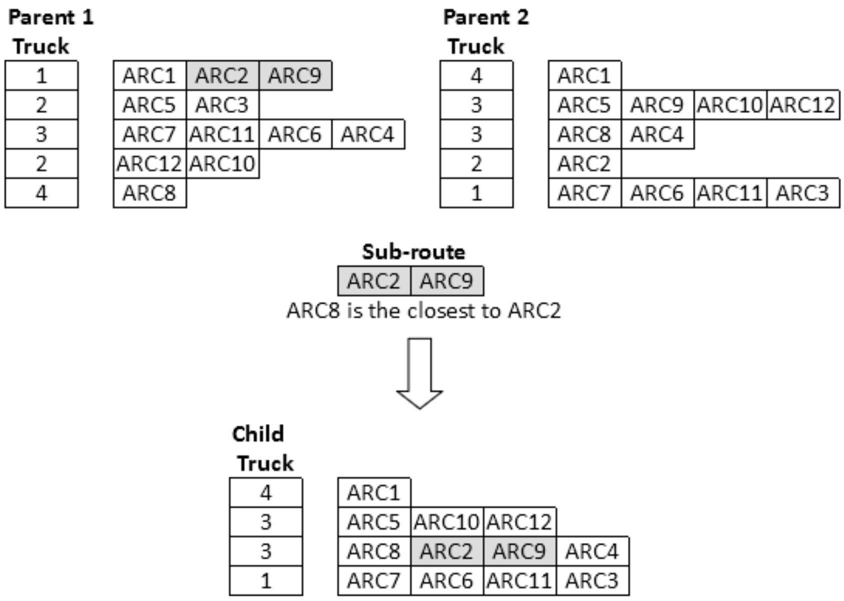

Figure 4.6 – Crossover operation

Capacities overflow can happen multiple times in this algorithm, to manage that, there is a list that stores every edge that could not be served for this reason. Once the previous process is complete, an iteration over the edges not served will take place assigning each edge to a truck as it is done in the initialization process of the GA. This guarantee that all edges are served by a truck.

4.3.4.3. Mutations

As the crossover, the mutation step must be specific for this problem being able to deal with the chromosome representation. The mutation is applied to every member of a new population with a low probability. This project implements two kinds of mutation, called swap mutation and inverse mutation, each of them have their own probability rate. Both mutation operators as the crossover can only generate valid solutions.

In the swap mutation, a truck is randomly select from the chromosome, then a random edge served by this truck in the solution is selected. The same occurs once more in the chromosome. Even if the edges are inside an array without the truck's array influence, is crucial to first select the truck so the capacity and the load of the truck are previously known for the mutation, this information is then utilized to only generate a valid solution. Having that two edges from the chromosome were randomly selected, a swap operation is done, the first edge takes the place of the second edge and the second edge is placed on the first's spot. The swap is done with validation to stop overflows from occurring. In the case of the validation fails, the swap process is repeated until the validation is successful, with a maximum of ten attempts. After that if the validation fails more than ten time, the mutation swap is not done.

The inverse mutation acts just in a sub-route of a truck in the solution. Like the sub-route selection in the crossover step, a sub-route is selected in this mutation. Then the order of the sub-route in inverted and inserted again in the path. This type of mutation differently of the swap mutation can not generate invalid solutions, as the number and weight of the ways are not changed and only the distance can change, if this GA had other constrains like the total length a truck can travel because of its fuel, this could generate an invalid solution and would require a validation, new constrains can be developed in future works and will be discussed in the future works section.

5. RESULTS AND DISCUSSION

6. CONCLUSIONS

7. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

1. Size of the streets that don’t allow big trucks to pass
2. Predict better the amount of waste in each street
3. Dijkstra's algorithm relies on the property that the shortest path from ss to tt is also the shortest path to any of the vertices along the path. This is exactly what BFS does.
4. Floyd-Warshall or djisktra

FUTURE:

- CONSTRAIN FOR THE AMOUNT OF FUEL A TRUCK HAVE AND HOW MUCH IT CAN TRAVEL UNTIL IT IS EMPTY

8. BIBLIOGRAPHY

OpenStreetMap contributors. (2015) Planet dump [Data file from $date of database dump$]. Retrieved from https://planet.openstreetmap.org.

Arakaki, R. K., & Usberti, F. L. (2018). Hybrid genetic algorithm for the open capacitated arc routing problem. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2017.09.020

Bătăgan, L. (2011). Smart Cities and Sustainability Models. *Revista de Informatică Economică*, *15*(3), 80–87. Retrieved from http://revistaie.ase.ro/content/59/07 - Batagan.pdf

Beliën, J., De Boeck, L., & Van Ackere, J. (2011). Municipal Solid Waste Collection and Management Problems: A Literature Review. *Transportation Science*. https://doi.org/10.1287/trsc.1120.0448

Buenrostro-Delgado, O., Ortega-Rodriguez, J. M., Clemitshaw, K. C., González-Razo, C., & Hernández-Paniagua, I. Y. (2015). Use of genetic algorithms to improve the solid waste collection service in an urban area. *Waste Management (New York, N.Y.)*. https://doi.org/10.1016/j.wasman.2015.03.026

Caragliu, A., del Bo, C., & Nijkamp, P. (2011). Smart cities in Europe. *Journal of Urban Technology*, *18*(2), 65–82. https://doi.org/10.1080/10630732.2011.601117

Carr, J. (2014). An Introduction to Genetic Algorithms. *Whitman College Mathematics Department*. https://doi.org/10.1016/S0898-1221(96)90227-8

Catania, V., & Ventura, D. (2014). An Approach for Monitoring and Smart Planning of Urban Solid Waste Management Using Smart-M3 Platform. *15th Conference of Open Innovations Association FRUCT*. https://doi.org/10.1109/FRUCT.2014.6872422

Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J. R., Mellouli, S., Nahon, K., … Scholl, H. J. (2012). Understanding smart cities: An integrative framework. In *Proceedings of the Annual Hawaii International Conference on System Sciences* (pp. 2289–2297). https://doi.org/10.1109/HICSS.2012.615

de Oliveira Simonetto, E., & Borenstein, D. (2007). A decision support system for the operational planning of solid waste collection. *Waste Management (New York, N.Y.)*. https://doi.org/10.1016/j.wasman.2006.06.012

Deng, X., Zhu, Z., Yang, Y., Li, X., Tian, Y., & Xia, M. (2007). A Genetic Algorithm for the Capacitated Arc Routing Problem. In *Proceedings of the IEEE: International Conference on Automation and Logistics*. https://doi.org/10.1145/1569901.1569947

Fadzli, M., Najwa, N., & Luis, M. (2015). Capacitated arc routing problem and its extensions in waste collection. In *AIP Conference Proceedings*. https://doi.org/10.1063/1.4915634

Fujdiak, R., Masek, P., Mlynek, P., Misurec, J., & Olshannikova, E. (2016). Using genetic algorithm for advanced municipal waste collection in Smart City. In *2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. https://doi.org/10.1109/CSNDSP.2016.7574016

Giffinger, R. (2007). Smart cities Ranking of European medium-sized cities. *October*, *16*(October), 13–18. https://doi.org/10.1016/S0264-2751(98)00050-X

Grimm, N. B., Faeth, S. H., Golubiewski, N. E., Redman, C. L., Wu, J., Bai, X., & Briggs, J. M. (2008). Global Change and the Ecology of Cities. *Science*, *319*(5864), 756–760. https://doi.org/10.1126/science.1150195

Guerrero, L. A., Maas, G., & Hogland, W. (2013). Solid waste management challenges for cities in developing countries. *Waste Management*. https://doi.org/10.1016/j.wasman.2012.09.008

HAN, H., & Cueto, E. P. (2015). Waste Collection Vehicle Routing Problem: A Literature Review. *PROMET - Traffic&Transportation*. https://doi.org/10.7307/ptt.v27i4.1616

Holland, J. H. (1992). Genetic Algorithms. *Scientific American*. https://doi.org/10.1038/scientificamerican0792-66

Hollands, R. G. (2008). Will the real smart city please stand up? *City*, *12*(3), 303–320. https://doi.org/10.1080/13604810802479126

Hoornweg, D., & Bhada-Tata, P. (2012). *What a waste: a global review of solid waste management*. *World Bank, Washington DC*.

Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, Adoption Barriers and Myths of Open Data and Open Government. *Information Systems Management*. https://doi.org/10.1080/10580530.2012.716740

Jebari, K., & Madiafi, M. (2013). Selection Methods for Genetic Algorithms. *International Journal of Emerging Sciences*.

Karadimas, N. V., Papatzelou, K., & Loumos, V. G. (2007). *Genetic algorithms for municipal solid waste collection and routing optimization*. *IFIP International Federation for Information Processing*. https://doi.org/10.1007/978-0-387-74161-1_24

Kumar, M., Husian, M., Upreti, N., & Gupta, D. (2010). Genetic Algorithm: Review and Application. *International Journal of Information Technology and Knowledge Management*.

Mendes, A. C. (2017). Almeida Henriques: “Smart cities” precisam de apoios. Retrieved June 1, 2018, from https://www.jornaldenegocios.pt/negocios-iniciativas/cidades-inteligentes/detalhe/almeida-henriques-smart-cities-precisam-de-apoios

Mitchell, M. (1995). Genetic algorithms: An overview. *Complexity*, *1*(1), 31–39. https://doi.org/10.1002/cplx.6130010108

Mohammed, M. A., Abd Ghani, M. K., Hamed, R. I., Mostafa, S. A., Ahmad, M. S., & Ibrahim, D. A. (2017). Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of Computational Science*. https://doi.org/10.1016/j.jocs.2017.04.003

Nam, T., & Pardo, T. A. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th Annual International Digital Government Research Conference on Digital Government Innovation in Challenging Times - dg.o ’11* (p. 282). https://doi.org/10.1145/2037556.2037602

Pereira, F. B., Tavares, J., Machado, P., & Costa, E. (2002). GVR: A New Genetic Representation for the Vehicle Routing Problem. *Artificial Intelligence and Cognitive Science: 13th Irish International Conference, AICS 2002, Limerick, Ireland, September 12-13, 2002 - Proceedings*. https://doi.org/10.1007/3-540-45750-X_12

Ramdane-Cherif, W. (2006). Evolutionary algorithms for capacitated arc routing problems with time windows. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*.

Saini, N. (2017). Review of Selection Methods in Genetic Algorithms. *International Journal Of Engineering And Computer Science*. https://doi.org/10.18535/ijecs/v6i12.04

Science for Environment Policy. (2015). Indicators for sustainable cities. *European Commission*, (12), 1–189. https://doi.org/10.2779/61700

United Nations. (2014). *World Urbanization Prospects 2014*. *Demographic Research*. https://doi.org/(ST/ESA/SER.A/366)

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*. https://doi.org/10.1007/BF00175354

Willemse, E. J., & Joubert, J. W. (2016). Constructive heuristics for the Mixed Capacity Arc Routing Problem under Time Restrictions with Intermediate Facilities. *Computers and Operations Research*. https://doi.org/10.1016/j.cor.2015.10.010

Wøhlk, S. (2008). A decade of Capacitated Arc Routing. *Operations Research/ Computer Science Interfaces Series*. https://doi.org/10.1007/978-0-387-77778-8_2

9. APPENDIX (OPTIONAL)

10.ANNEXES (OPTIONAL)