FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Route Optimization for the Solid Waste Collection

Mariana Soares Maciel Machado Magalhães

WORKING VERSION

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: José António Soeiro Ferreira (PhD)

Co-orientador: Ana Maria Rodrigues (MSc)

January 30, 2012

Resumo

O aumento da população registado nos últimos anos tem conduzido a um aumento da produção de resíduos sólidos urbanos. Este aumento traduz-se numa maior degradação das condições ambientais tornando assim a gestão de resíduos num processo cada vez mais crucial para garantir uma boa qualidade de vida à população.

O processo de gestão deste tipo de resíduos irá ser constituído por diversas etapas: recolha, transporte e depósito em destino final. O trabalho realizado ao longo desta dissertação centra-se na construção de diversas rotas percorridas por camiões destinados à recolha deste tipo de resíduos.

O principal objetivo que se pretende alcançar com este trabalho centra-se na otimização das rotas percorridas pelos diversos camiões de recolha.

A otimização de rotas em problemas de transporte poderá ser abordada de diferentes formas. Nesta área poderão ser encontrados diversos trabalhos realizados por diferentes autores com a utilização do mais variado tipo de métodos. A escolha da metodologia a aplicar irá variar muito de problema para problema, sendo necessária a análise das diversas características deste para a escolha acertada da abordagem a seguir.

Ao longo desta dissertação, tendo em vista a obtenção de resultados satisfatórios, foi aplicada uma metodologia composta por três fases distintas.

Na primeira fase desta abordagem são utilizados programas de otimização, tendo em vista a obtenção de uma solução inviável. Esta será composta por apenas uma rota, não cumprindo a restrição associada ao limite máximo da capacidade que dado veículo poderá transportar.

A solução obtida na fase anterior – Mega-Rota – será utilizada pela segunda fase da abordagem utilizada. Nesta fase serão utilizados métodos heurísticos com vista a obtenção de uma solução viável que já terá em conta a restrição associada às capacidades de transporte dos diversos veículos disponíveis.

Por fim, na terceira fase desta metodologia será aplicada uma Metaheurística, designada por Pesquisa Tabu, que tem como objetivo a melhoria da solução obtida na fase anterior.

A metodologia aplicada será validada através de testes realizados a diversas instâncias de um dado problema já estudado por outros autores. Para a avaliação dos resultados obtidos, estes foram comparados com os diferentes resultados alcançados para o mesmo problema retirados da literatura.

Abstract

The growth of the population takes us to a highest production of municipal solid waste. This increase leads to a higher degradation of the environmental conditions making the solid waste management a crucial process to provide a good quality life to the population.

The solid waste management is composed by a certain number of processes, like collection, transportation and deposit on final destination. The work developed during this Master Thesis focuses in the construction of the routes travelled by the vehicles destined to realise the collection of this type of refuse.

The main purpose of the work developed during this Master Thesis is to optimize the routes travelled by different trucks assigned to realize the collection service.

The route optimization in transportation problems can be approached by several different ways. In this field can be found a several number of different works developed by different authors using many distinct approaches. The methodology chosen to solve a certain problem will vary a lot from problem to problem, being required to analyse the characteristics of the problem under concern.

During this Master Thesis was applied a methodology composed by three distinct phases with the purpose of obtaining satisfactory results to the tested instances.

In the first phase of the methodology are used optimization programs that aims to obtain an infeasible solution. This solution, called Mega-Route, is composed by one route that doesn't take into account the vehicle capacity constraint.

The solution obtained in the previous phase is used as an input in the second phase of this approach. In this phase are used Heuristic Methods in order to obtain a feasible solution that uses the restriction on the total amount of Urban Solid Waste collected by a vehicle.

Finally in the last phase of this methodology a Metaheuristic is applied, known as Tabu Search, which has the purpose of obtaining a better solution than the one obtained in the previous phase. This solution should be close to the optimal one.

The validation of the methodology occurs through testing a several number of instances that composes a given problem already studied by other authors. To evaluate the obtained results, these are compared with the results attained by other authors.

Agradecimentos

A realização deste trabalho não seria possível sem a ajuda e amizade de diversas pessoas direta ou indiretamente ligadas a esta dissertação.

Gostaria de agradecer aos meus orientadores, José Ferreira Soeiro e Ana Rodrigues, por toda a ajuda, disponibilidade, conselhos e confiança depositada para a realização desta dissertação.

Não poderia deixar de agradecer à Teresa Bianchi pela ajuda prestada na fase inicial deste trabalho sem a qual teria sido muito mais complicado a entrega desta dissertação.

Ao Neeb e ao Pantufas não só por todos os momentos de descontração e de mais seriedade partilhados durante este último semestre, mas também por todos os momentos partilhados ao longo destes últimos anos.

À Cris, ao Tiago Santos, ao Half, à Cátia e ao Alexandre. Obrigada pela vossa disponibilidade, pela seriedade quando as situações o exigiam e por toda a ajuda disponibilizada.

Ao Hugo por ter sido a pessoa que mais conviveu comigo durante este semestre. Obrigada por todos os momentos de descontração, pela ajuda mas acima de tudo pela enorme paciência.

A uma pessoa muito importante para mim, obrigado Pedro por todos os momentos partilhados, todo o carinho e paciência pois sei que nem sempre foi fácil lidar comigo principalmente durante a fase final desta dissertação. Agradeço toda a disponibilidade, ajuda e todos os conselhos dados nos momentos certos. Mas acima de tudo, por nunca ter duvidado de mim.

A duas pessoas também muito importantes. À Sofia, por ser o meu porto de abrigo quando eu mais precisei e por me ter apoiado sempre, mesmo em situações mais delicadas. Ao Tiago, por me ter acompanhado desde o primeiro até ao último ano de curso nos momentos bons e nos menos bons. A vocês, o meu mais sincero obrigado e espero, sinceramente, que a nossa amizade perdure por muitos anos.

Por último, mas mais importante, gostaria de agradecer à minha família. Aos meus tios e primos por toda a boa disposição partilhada. Aos meus avós por, durante toda a vida, terem feito os possíveis e impossíveis por mim. Ao meu avô, que apesar de já não estar entre nós, sempre foi e é um exemplo para mim e por saber que seria um orgulho enorme ele ver-me terminar esta fase da minha vida. Gostaria de agradecer em especial aos meus pais, por serem os meus maiores exemplos de compreensão, carinho, persistência, coragem e amizade. Por sempre terem investido em mim e por nunca terem duvidado das minhas capacidades. O meu muito obrigado por toda a paciência, em especial na última fase da realização desta dissertação.

A todos, o meu muito obrigado.

"Logic will get you from A to B.
Imagination will take you everywhere."

Albert Einstein

Contents

List of Figures

List of Tables

Abbreviations

ARP	Arc Routing Problem
CARPIF	Capacitated Arc Routing Problem with Intermediate Facilities
CARP-RP	Capacitated Arc Routing Problem with Refill Points
CARPUD	Capacitated Arc Routing Problem with Unit Demand
CPP	Chinese Postman Problem
CPU	Central Processing Unit
CVRP	Capacitated Vehicle Routing Problem
DRPP	Directed Rural Postman Problem
DyRPP	Dynamic Rural Postman Problem
ECARP	Extended Capacitated Arc Routing Problem
GLPK	Gnu Linear Programming Kit
GRP	General Routing Problem
HCPP	Hierarchical Chinese Postman Problem
HRPP	Hierarchical Rural Postman Problem
LARP	Location Arc Routing Problem
lp	Linear Programming
LRPP	Location Rural Postman Problem
MDVRP	Multiple-Depot Vehicle Routing Problem
MDVRPWRC	Multiple-Depot Vehicle Routing Problem with Wheight-Related Costs
m-HRPP	Multiple Hierarchical Rural Postman Problem
mod	Model
MRPP	Mixed Rural Postman Problem
m-RPP	Multiple Rural Postman Problem
m-TSP	Multiple Travelling Salesman Problem
NN	Nearest Neighbour
NRP	Node Routing Problem
OM	Optimization Model
PCARP	Periodic Capacitated Arc Routing Problem
PdRPP	Periodic Rural Postman Problem
PERSU	Plano Estratégico para os Resíduos Sólidos Urbanos
PRPP	Privatized Rural Postman Problem
RPP	Rural Postman Problem
SARP	Sectoring Arc Routing Problem

SARP	Stochastic Arc Routing Problem
USW	Urban Solid Waste
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows
WGRP	Windy General Routing Problem
WRPP	Windy Rural Postman Problem
WRPPZ	Windy Rural Postman Problem with ZigZag Service
XPP	Crossing Postman Problem

Chapter 1

Introduction

1.1 Motivation

The growth of the population takes us to a highest production of municipal solid waste. In 2005, as we can see in Figure 1.1, the production of solid waste in Portugal reached 4.5 millions of tons, namely 1.24 Kg by day for an inhabitant. This increase will lead to a degradation of environmental conditions. To tackle this degradation, PERSU (*Plano Estratégico para os Resíduos Sólidos Urbanos*) was approved by the government in 1997. Having the definition of new strategies, goals and priorities between 2007 and 2016 as objective, the PERSU II was later approved. [9]

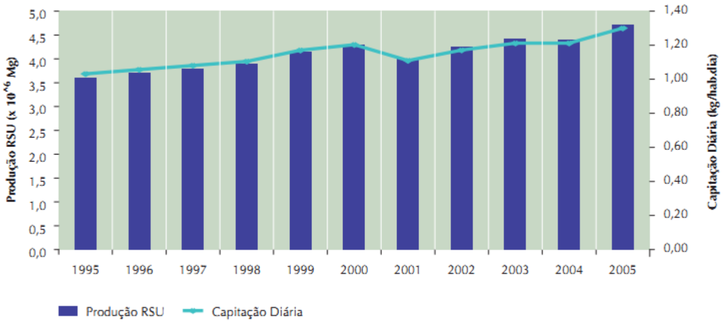


Figure 1.1: Solid Waste production in Portugal [9]

Due to the importance of solid waste management, its collection is absolutely essential to the correct functioning of the whole process. The characteristics of the collection process will vary from city to city. In each one, there are multiple companies responsible for the management of all the waste produced by their inhabitants. The aim of these companies is to reduce their costs to the minimum, like every company that is on the market. A convenient selection of the routes to be traversed by their vehicles, for collection of municipal solid waste, certainly is one of the relevant areas for costs reduction.

The work developed during this Master Thesis is directly connected to these costs reduction. The main objective of this work is to contribute to the optimization of the routes travelled by the different trucks, taking into account different restrictions due to different characteristics of the situation under concern.

A particular problem will be described, inspired on real situations, which takes into account a group of important restrictions. Solutions methods will be described and various instances of the problem will be discussed and solved.

Besides their practical relevance, it should be mentioned that this type of problems are complex and challenge modelling and optimization procedures.

Based on the needs presented above, we expect that the work developed during this Master Thesis may contribute, directly and indirectly, to deal with new optimization problems and to the companies of this sector. In fact, many of these firms don't have computational solutions able to handle the real problems they face everyday.

1.2 Problem Description

The proposed problem, based on [2] [18], consists in optimizing the routes that have to be traversed by different vehicles, through the use of new methods that will be explained.

This problem will be formulated using Graph and Networks Theory:

- Arcs will represent several streets from a city and can be of two different types:
  - Required arcs: this set contains all the streets that need to be serviced by the vehicles.
  - Arcs not required: set of arcs that will be crossed only when it's necessary to reach the required ones.

    And to the existing arcs will be associated two different costs:
    * Service Cost: Cost activated when a truck realizes service on the correspondent arc. This cost is associated only to the required arcs.
    * Deadheading Cost: Cost activated when a vehicle transverse an arc without servicing it.
- Nodes will represent the street crossings or dead-end streets.

There is another cost not mentioned yet, named disposal cost, and it's activated when the vehicle returns to the depot to dispose all the refuse collected until that moment.

Besides these characteristics already mentioned, there are constraints that have to be taken into consideration. The principal ones are:

- Capacity Constraint: Every vehicle has limitations on their transport capacity.
- Arc Constraint: The direction of the different arcs has to be the correct one.

There are more constraints that will be explained later, this two are the most important to comprehend how the studied problem is defined by the author.

1.3 Methodology

The methodology adopted in this Master Thesis can be seen in Figure 1.2. As we can see, it is composed by three different phases:

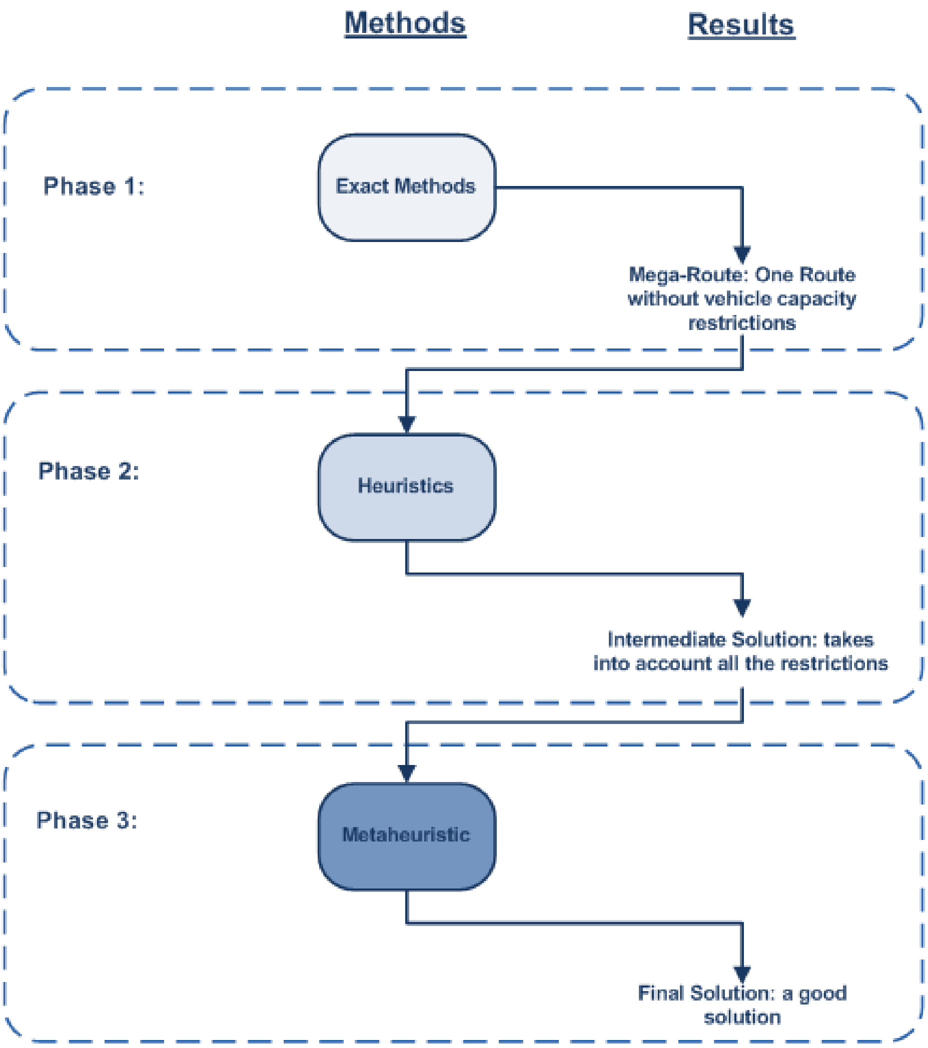


Figure 1.2: Methodology

- Phase 1 – In the first phase of the methodology are used optimization programs to find a solution, although an infeasible one for the main problem. The capacity of the vehicles is not considered in this phase, so the solution obtained is composed by just one route with no limitations on the quantity of urban solid waste collected.
- Phase 2 – In this phase, are used Constructive Heuristics to find the first feasible solution. So, we take the previous solution obtained, and we construct a new solution taking into account the capacity of different trucks. This solution contains so many routes as the number of available vehicles.

- Phase 3 – During this phase a Metaheuristic approach will be followed. Tabu Search will be applied to the solution obtained in the previous phase. The purpose of this phase is to find a feasible solution with an objective function's value close to the optimal one.

1.4 Master Thesis Structure

The Master Thesis is composed by 6 chapters. The first one, chapter 1, includes the motivation for the subject-matter and the first explanations how the selected problem will be solved.

Chapter 2 contains the basic concepts to the full comprehension of the subject studied on this Master Thesis.

Chapter 3, named Methods of Optimization contains a brief introduction to some Methods capable of solving optimization problems.

Chapter 4 intends to show a set of relevant Optimization Models and the Methodology adopted to solve the problem under concern.

Chapter 5 contains all the results obtained to the different instances of the problem and the discussion of these results.

Finally, chapter 6 contains the conclusions drawn from the Master Thesis and the work that can be accomplished in the future.

All the chapters contain a brief introduction explaining the principal subjects that will be exposed in them and a final summary of the respective chapter's content.

Chapter 2

Urban Solid Waste Collection

This chapter is divided into four different sections. The first one contains information about how the USW (Urban Solid Waste) can be collected.

The second and the third sections of this chapter contain the key concepts to the comprehension of the assumptions considered to the resolution of the problem studied on this Master Thesis. First, can be found all the fundamental definitions related to the Graphs and Networks Theory. And after the explanation of these concepts, we expose some approaches to the main problems studied on Solid Waste Collection field: TSP (Travelling Salesman Problem) and RPP (Rural Postman Problem).

2.1 Urban Solid Waste

Definition of Urban Solid Waste:

" The domestic wastes or some other similar wastes, according to their nature or composition, namely those coming from the tertiary sector or from the commercial, trading or industrial plants and from health care units, providing that, in any case, the daily production do not exceed 1100 L per producer " - [26]

How the urban solid waste is collected

The municipal solid waste collection involves three basic phases: collection, transport and deposit on final destination.[10] This Master Thesis will focuses on the transport phase.

The USW can be collected in two different ways:

- Mixed waste collection: Waste is collected indiscriminately by vehicles that could have different capacities. This is the type of collection that is implemented in most cities. The trucks (Figure 2.1a) used to accomplish it are the most common ones, such as the containers (Figure 2.1b) used by the USW producers .
- Separate waste collection: Waste is collected separately by vehicles different from those used in mixed collection (Figure 2.2a). In most cases, this type of collection will focus on

(a) Vehicle (b) Container

Figure 2.1: Mixed waste collection

the ecopoints, which are a set of containers that contain different bins to different types of garbage (Figure 2.2b).

(a) Vehicle (b) Ecopoint

Figure 2.2: Separate waste collection

In both cases, the respective vehicles and containers could have different sizes. This will depend on where they are located.

Each one of the selected vehicles should follow a previously established route, aiming the collection of a stipulated amount of garbage. These trucks will start and end its routes on a facility, called depot.

There are some characteristics that will vary from problem to problem. The most important is the limitation on the amount of the USW that each vehicle can transport, which is known as vehicle capacity.

Depending on the location, the fleet of trucks can be of two different types:

- Homogeneous: composed by vehicles that have the same characteristics.
- Heterogeneous: constituted by vehicles that have different characteristics.

Every streets that need to be serviced, will be transverse by one and only one vehicle responsible to realize the collection of USW.

2.2 Graphs and Networks Theory

This section contains some basic concepts of Graphs and Networks considered relevant to the full comprehension of the subject studied on this Master Thesis. [24]

2.2.1 Graphs and Networks

A graph G consists on several pairs of (V, E), where V symbolizes the vertices (nodes) and E the edges/arcs of a problem. Each arc will connect one pair of vertices. There are two different types of graphs:

- Oriented Graphs: Consists on pairs of arranged vertices - Figure 2.3

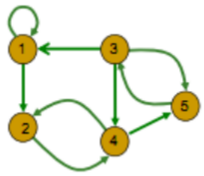


Figure 2.3: Oriented Graph [15]

- Non-oriented Graphs: Consists on pairs of non-arranged vertices - Figure 2.4

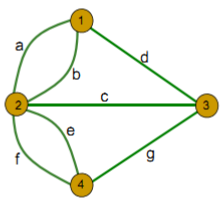


Figure 2.4: Non-oriented Graph [15]

A network can be defined as a graph where to each vertex, a particular value is associated that can be related to several characteristics, as distance, cost, capacity, among others. These different characteristics depend on the problem under study (Figure 2.5).

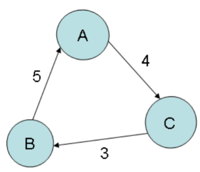


Figure 2.5: Network [15]

2.2.2 Connected Graph

Defined as a graph where there always exists a path between any two vertices belonging to this graph.

2.2.3 Path and Circuit

- Path: sequence of vertices, where each one is connected to the next one of the sequence. The first one is called initial vertex and the last one, final vertex – Figure 2.6.

Figure 2.6: Path

- Circuit: set of vertices that begins and ends on the same vertex. – Figure 2.7.

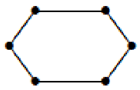


Figure 2.7: Circuit

2.2.4 Vertex Degree

A vertex degree is equal to the number of the edges that are linked to it.

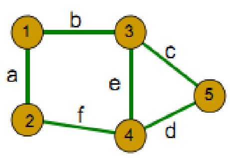


Figure 2.8: Vertex Degree [15]

In Figure 2.8:

deg(1)=2;deg(2)=2;deg(3)=3;deg(4)=3;deg(5)=2;deg(6)=2;deg(7)=2;deg(8)=2 - where deg(x) represents the degree of vertex x.

2.2.5 Eulerian Graph

- Eulerian Circuit – Circuit that contains all the edges from a graph.

- Eulerian Graph – Graph that contains at least one Eulerian Circuit.

Theorem 1: A connected graph contains one Euler path, if and only if, every vertex from this graph has an even degree.

2.3 Relevant Problems to Urban Solid Waste Collection

There are two main approaches used to solve the Solid Waste Collection problems: one is based on ARP (Arc Routing Problems) and the other in NRP (Node Routing Problems).

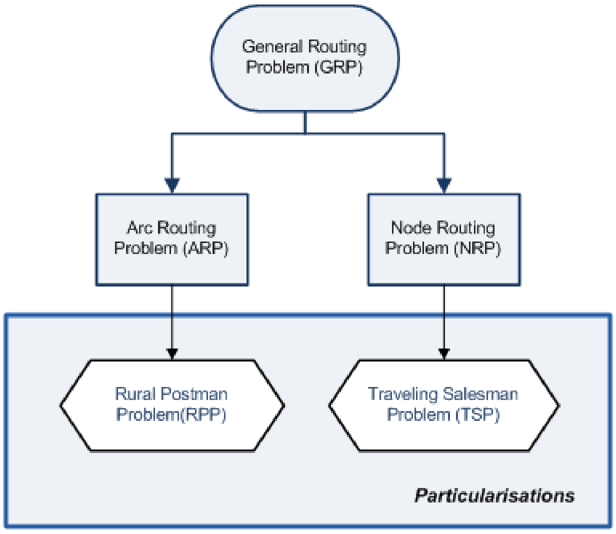


Figure 2.9: General Routing Problems Approaches

These approaches are illustrated in Figure 2.9.

- Arc Routing Problems (ARP): The aim of this type of problem is to find a set of routes constituted by the arcs/edges of a graph, satisfying different conditions. [6, 13, 14, 12]
- Node Routing Problems (NRP): The aim of this type of problem is to find a set of routes constituted by the nodes of a graph satisfying different conditions. [3, 17]

Furthermore, there are some particularisations of the approaches explained before. Below, in Figure 2.9, are explained two of them due to their relevance to the Solid Waste Collection problems: TSP and RPP.

2.3.1 TSP (Travelling Salesman Problem)

The Travelling Salesman Problem is a well-known problem in Operations Research field. Its importance lies on the difficulty to find the optimal solution of the certain problems due to its computational complexity and on the large range of applications that it can be used.

This type of problem is composed by a certain number of cities. The salesman has to visit all the cities of the problem and return to the initial city visited. So the solution ends where it begins. The aim of solving these problems is to find the minimum distance that the salesman needs to travel. [5]

Next, will be presented two different particularisations of the TSP (Figure 2.10):

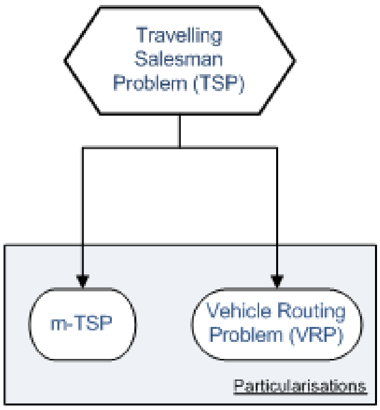


Figure 2.10: Travelling Salesman Problem Particularisations

1. VRP (Vehicle Routing Problem) – The VRP is a new extension of TSP, but to all the nodes of the problem is associated a demand value. The aim is to find a set of routes travelled by a set of vehicles. All the vehicles need to begin and end at the same node, named depot. The objective is that all the customers are supplied with their demands and the travelled distance by all the vehicles is minimized. [28] There are some particularisations of the basic VRP [28, 16, 17]:

    - SPVRPTW (Stochastic Periodic Vehicle Routing Problem with Time Windows)
    - SVRP (Stochastic VRP)
    - PVRP (Periodic VRP)
    - VRPTW (VRP with Time Windows)
    - MDVRP (Multiple-Depot VRP)
    - MDVRPWRC (Multiple-Depot VRP with Weight-Related Costs)
    - CVRP (Capacitated VRP)

2. mTSP (Multiple-Travelling Salesman Problem) – The target is to find a set of routes travelled by a certain number (m) of salesmen. All the routes have to begin and end on the same city. All the cities are visited only once. [1]

    All the VRP approaches mentioned previously can be considered to the mTSP, since these kind of problems are considered a relaxation of the VRP. [1]

2.3.2 RPP (Rural Postman Problem)

The RPP was first proposed in 1974 defined as the problem of traversing each required arc of a network at least once at minimum cost. [6, 30] The RPP has 3 different variants that can be observed on Figure 2.11: [30]

- DRPP (Directed Rural Postman Problem) – Problem that is composed by graphs containing directed arcs.
- URPP (Undirected Rural Postman Problem) – This kind of problem consists on graphs with non-directed arcs.
- MRPP (Mixed Rural Postman Problem) – The MRPP is composed by graphs with two types of arcs: directed and undirected.

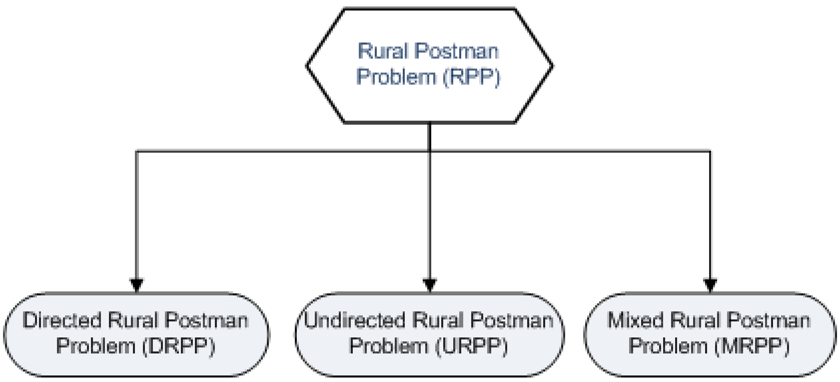


Figure 2.11: RPP

Then, two particularisations of the RPP will be presented [30]:

1. CPP (Chinese Postman Problem) – The aim of the CPP is to find the minimum path travelled by a postman in an undirected graph.
2. SCP (Stacker Crane Problem) – This kind of problem can be related directly to the RPP, but have some different conditions. We can place the SCP into the MRPP type of problem. The aim is to find the shortest circuit which will contain all the arcs of the problem at least once.

As we can see in Figure 2.12 there are some generalisations of this type of problems. Some of them are exposed next. [30]

1. Windy Problems – This type of problem, as can be seen on Figure 2.12, is a generalisation of the RPP. In it, the cost of traversing an edge can be different depending on the direction that the edge is traversed.

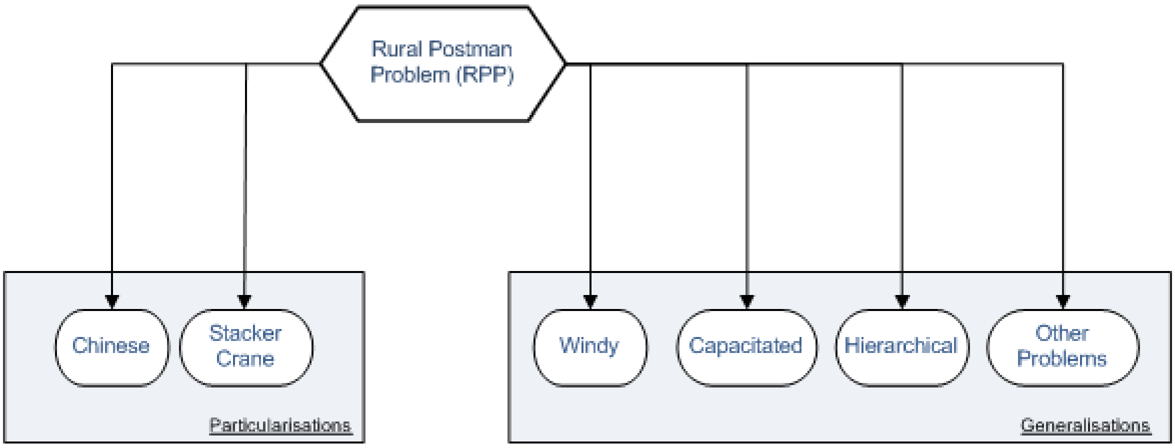


Figure 2.12: RPP particularisations and generalisations

- WRPP (Windy RPP)
- WRPPZ (Windy RPP with Zigzag Service)
- WGRP (Windy GRP)

2. Capacitated Problems – Problems composed by a graph where to the required arcs are associated different demands that have to be collected by different vehicles, each one with a maximum transport capacity.

   - PCARP (Periodic Capacitated Arc Routing Problem)
   - LARP (Location ARP)
   - CARP – RP (CARP with Refill Points)
   - SARP (Sectoring ARP)
   - SARP (Stochastic ARP)
   - CARP with time dependent service costs
   - ECARP (Extended CARP)
   - CARPIF (CARP with Intermediate Facilities)
   - CARPUD (CARP with Unit Demand)
   - Multi-objective version of the CARP

   The problem solved in this Master Thesis will be a Capacitated Problem, as can be seen on chapter 4. So, this generalisation will be further explained on this chapter.

3. Hierarchical Problems – On this type of problems the different conditions are related to the order in which the edges are crossed. There are a lot of variations of this type of problems, like:

- HRPP (Hierarchical RPP)
- m-HRPP (Multiple Hierarchical RPP)
- HCPP (Hierarchical CPP)

4. Other Problems – There are some other problems that comprise different restrictions from the ones explained before. Some of them are:

   - m-RPP (Multiple Vehicle RPP)
   - PdRPP (Periodic RPP)
   - SMTPP (Scheduled Multiply Traversed Postman Problem)
   - PRPP (Privatized RPP)
   - XPP (Crossing Postman Problem)
   - DyRPP (Dynamic RPP)
   - LRPP (Location RPP)

2.4 Summary

The Chapter intends to provide a brief introduction to the subject-matter studied during this Master Thesis.

There are two different ways of collecting the USW – Mixed and Separate Waste Collection - and the resources used by these two types of collection will be different. The characteristics of the vehicles assigned to realize this service could vary from local to local, as well as the containers. The fleet of trucks can be for two different types: Homogeneous or Heterogeneous.

After the explanation of the real life concepts associated to this type of problems, the author exposes a set of basic concepts about Graphs and Networks Theory.

Finally, it's done a brief introduction to relevant problems to USW. This type of problems can be divided into two fundamental approaches: ARP and NRP.

When referring to the ARP this chapter includes information about the RPP particularisation. This type of problem has three different variants: DRPP, URPP and MRPP and has some particularisations too. The RPP particularisation's here explained are: CPP and SCP.

The other approach presented in this chapter is TSP that has two different particularisations: m-TSP and VRP that are exposed too.

After reading the present chapter, reader should be able to comprehend the real and theoretical concepts associated to the collection of this type of refuse.

Chapter 3

Methods of Optimization

To solve optimization problems there are different kinds of approaches that can be used. The selection of which approach must be followed depends, in general, of different aspects and, of the particularities of the problem under study. A particular method can be a good choice to solve a specific problem, but a bad choice to other one. Due to this fact, the choice of which method should be used is a crucial step to find good final solutions.

This chapter is divided into three main sections that contain different approaches used to solve optimization problems.

The first one is composed by a group of Exact Methods that could provide optimal solution to a given problem. The last two sections contain some concepts about Heuristics and Metaheuristics, approximative methods, that are used to find good solutions, but not necessarily the optimal ones.

The USW (Urban Solid Waste) collection will be inserted in the optimization problems field. This type of problems can be defined as: [20]

$opt\{C(x)|x \in S\}$, where S is defined as a discrete and finite set. The main purpose of this type of problems is to find the optimal solution $x^* \in S$, in other words:

$$\forall_{x \in S} \quad C(x^*) \leq C(x)$$

3.1 Exact Methods

This section provides a brief introduction to two general fundamental exact methods: Branch and Bound and Branch and Cut. Moreover a particular exact technique, the Dijkstra Algorithm, will also be introduced due to its relevance to the case of this thesis.

In spite of this chapter being divided in three distinct sections, the first two exact methods should be separated from the last one.

Branch and Bound and Branch and Cut are used to solve complex optimization problems and will provide the optimal solution to these. When a problem is too complex these two methods demand a lot of computational effort and may not be able to provide the optimal solution.

The last technique, Dijkstra Algorithm, is used to obtain the shortest path between two chosen nodes. In this thesis is used to complete different routes when this is necessary.

3.1.1 Branch and Bound

This method appears in 1960 proposed by Lang and Doig that were solving linear problems with integer decision variables. The first time that this approach was applied to TSP (Travelling Salesman Problem) was in 1962 by Murty, Karel and Little. [19, 8]

Most of the combinatorial problems have many feasible solutions. The principal purpose of this method is to partioning the set of feasible solutions into smaller sets becoming easier to find the optimal solution. [8]

Although this method is able to provide the optimal solution, it should be applied with caution to some combinatorial problems. If the problem is too complex will demand a high computational effort and, as mentioned before, this is not advised. [19]

3.1.2 Branch and Cut

This method can be defined as a particularization of the Branch and Bound method.

The Branch and Cut approach uses Branch-and-Bound algorithm combined with the use of a cutting plane method. This combination will improve the relaxation of the problem and uses the divide and conquer approach to solve problems. [23]

The principal difficult of this method lies on the decision of how to cut and how to generate cutting planes. [23]

3.1.3 Dijkstra Algorithm

This algorithm was published by Edsger W. Dijkstra in 1959 and it is one of the most popular algorithms used in Operational Research field. [19]

This method differs from the other exact methods because is used just to find the shortest path between two chosen nodes. So, this approach is only used in simple problems with no constraints attached to them. This minimum path is calculated taking into account different costs or distances of transverse the different arcs of a problem.

In this Master Thesis, the Dijkstra Algorithm is used to calculate the minimum distance between the last arc serviced by a truck and the next one. In chapter 4 can be found a more detailed explanation of this algorithm.

3.2 Heuristics

Unlike the Exact Methods, through the application of a Heuristic the solution obtained may or may not be the optimal one. To verify the quality of a heuristic we need to check two main characteristics: quality solution and computer time. When applying a heuristic we pretend to obtain a solution near to the optimal in the shortest possible time. [32]

In this section we can find a brief summary to the main concepts of some well-known groups of heuristics:

- Constructive Heuristics;
- Improvement Heuristics;
- Compound Heuristics.

The aim of this part of the chapter is to provide an introduction to the main principles of heuristics when applied to the TSP and CPP (Chinese Postman Problem), rather than to give a detailed explanation of the existing heuristic algorithms.

3.2.1 Constructive Heuristics

The main purpose of these heuristics is to construct a solution by adding distinct points of the problem to the solution. When every points of the problem are in the solution, the method ends and the final solution is attained.

The way that the points are added to the solution under construction will depend on the heuristic used.

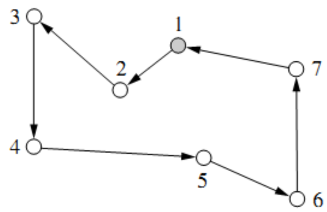
Bellow, are presented the basic concepts of two different constructive heuristics:

Nearest Neighbour (NN)

In this heuristic the various points of a problem will be added to the solution taking into consideration the criterion of the closest neighbour. [32]

	1	2	3	4	5	6	7
1	0	75	135	165	135	180	90
2	75	0	90	105	135	210	150
3	135	90	0	150	210	300	210
4	165	105	150	0	135	210	210
5	135	135	210	135	0	90	105
6	180	210	300	210	90	0	120
7	90	150	210	210	105	120	0

(a) Data of the problem



(b) Final Solution

Figure 3.1: NN heuristic [32]

The Figure 3.1 shows an example of the application of this heuristic. In it we can see the data of the problem (Figure 3.1a) and a tour constructed through the use of NN heuristic. (Figure 3.1b):

- First Step: Choose point 1 as the first node.
- Second Step: Add the point nearest to the last one added. This step is repeated until all of the points are present on the solution.
- Third Step: Connect last point to the first one.

Clarke-Wright

This heuristic appears in 1964 proposed by Clarke and Wright that was solving a CVRP (Capacitated Vehicle Routing Problem) and it's known also by Savings Heuristic. [4]

The concept used on this heuristic is based on the *savings* obtained by combining two small routes into a larger one. The saving is calculated through the application of the following equation [4]:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

To use this heuristic, there are different steps that must be followed:

- Step 1: Each route begins with one customer.
- Step 2: All the savings are calculated and sorted from largest to smallest.
- Step 3: At each iteration the next saving is considered and if the two associated customers can be feasibly merged into a new route, then the routes are merged.

The main advantages of this algorithm are its easy implementation and its speed.

3.2.2 Improvement Heuristics

The main purpose of implementing this type of method is to improve a solution already found. Due to this objective, before we apply it, we need to build a solution using another approach.

The way in which the solution initially found will be improved depends on the particularities of the improvement heuristic used. Bellow, are presented the basic concepts of two improvement heuristics (Exchange Heuristic and Local Search Heuristic).

Exchange Heuristic

This heuristic begins with a feasible solution and intends to find a better one. This final solution will be found through the specific exchange of the first solution that is going to be break in k links in a systematic way. Then, the paths are going to be joined and the solution obtained is compared with the best solution found until that moment. [32]

In Figure 3.2 an example of an application of this heuristic can be observed. First we have an initial solution that is going to experience some changes. The final solution that has to be better than the first one, is obtained through the exchange of some arcs of the problem. The example presented on this figure is an implementation of the 2-opt heuristic, because exchanges occurs between two arcs of the respective problem.

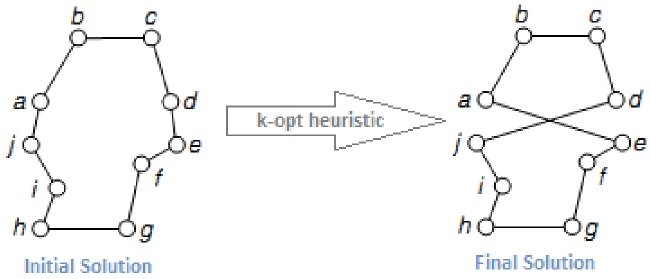


Figure 3.2: K-opt Heuristic

Local Search

The main objective of this kind of heuristic is to explore the neighbourhood of a given solution and find a better one.

This method will be repeated as many times as necessary taking into account a stopping criterion. This criterion can be the number of iterations, the number of iterations without find a better solution, among others. [27]

3.2.3 Compound Heuristics

This type of heuristics is composed by all the methods that enclose more than one phase to obtain a final solution.

In Figure 3.3 can be seen an example of a Compound Heuristic. As shown in the aforementioned figure, in the first phase a constructive heuristic is applied to obtain an initial solution and finally is used an improvement heuristic to find the final solution that should be better than the initial one.

3.3 Metaheuristics

Unlike many heuristic techniques mentioned before that solves only a specific problem, Metaheuristics are able to solve different combinatorial optimization problems. [32]

These methods appear in the 80s with the purpose of solving hard optimization problems as well as possible. The most relevant characteristic of these techniques are the analogies to a large number of subjects like physics, biology or ethology. [32, 11]

Like other methods mentioned before, Metaheuristics have advantages and disadvantages. The main advantage associated to these methods is the ability to solve different types of problems. The main disadvantages are the parameter adjustment and the large computational time needed. [32]

On this section we find four different Metaheuristics that were chosen due to their importance on the area of optimization problems and to provide a large spectrum of analogies. The methods exposed in this section are:

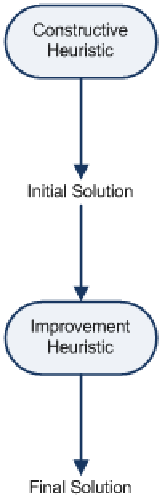


Figure 3.3: Compound Heuristic

- Simulated Annealing
- Genetic Algorithms
- Ant Colony Algorithms
- Tabu Search

These different methods are based in two different techniques: [32]

- Local Search Techniques: aims finding a better solution on the space neighbourhood of the current solution.
- Population Search Techniques: a population of solutions are combined with the purpose of create new generations constituted by better solutions that the previous population.

Nowadays, hybrid methods are also used. They expect to gain from the combination or integration of parts of different Metaheuristics and/or Exact Methods. [11, 22]

3.3.1 Simulated Annealing

The Simulated Annealing Metaheuristic is based on a physical system, as can be seen on Table 3.1.

Table 3.1: Analogy between an Optimization Problem and a Physical System [11]

Optimization Problem	**Physical System**
Objective function	Free energy
parameters of the problem	"coordenates" of the particles
find a "good" configuration (even optimal configuration)	find the low energy states

The main objective of this process is to decrease slowly the temperature. In the optimization problems, this Metaheuristic aims the reduction of the value of the objective function. This purpose is achieved through the variation of a parameter, $\triangle$E. When:

- $\triangle$E <0: new configuration has a lower energy.
- $\triangle$E >0: new configuration has a higher energy. However, this configuration may not be automatically excluded.

A worst solution is accepted as a new initial configuration if the probability given by $e^{\frac{-\Delta E}{T}}$ is higher than a number between [0, 1] generated randomly. [11, 32]

The principal difference between simulated annealing and local search is the possibility to accept non-improved solutions. This fact allows the exploration of different regions of the feasible space. [32, 11]

3.3.2 Genetic Algorithms

The Genetic Algorithms are based on the Darwin's Theory of evolution of the species and are used to solve complex combinatorial optimization problems. [11, 32]

This method is constituted by four basic steps: [32]

- First step: Generate a set of solutions, called Population.
- Second step: Evaluation of the solutions generated in the previous step.
- Third step: Good solutions are considered and the remaining ones are eliminated.
- Fourth step: The good solutions go through processes like crossover, reproduction or mutation and a new generation is created.

It's expected that the new generations will be better than the previous ones. This method will stop when the stop condition is reached. [32, 11]

Like the Darwin's Theory aims survival of the fittest individual, this Metaheuristic have the purpose of finding good final solutions.

3.3.3 Ant Colony Algorithms

These algorithms were proposed by Colorni, Dorigo and Manniezzo in 1992. [11]

The Ant Colony Algorithms are relevant to solve some optimization problems by the simulation of the ant's colony behaviour. These insects always follow the same track to find food and this path is the shortest one. The communication between this type of insects is done through a chemical substance, called pheromone. [11]

This Metaheuristic comprises 3 different phases [21]:

- Phase 1: Create a certain number of solutions by the ants.
- Phase 2: The solutions created in Phase 1 are improved through the use of a Local Search Heuristic.
- Phase 3: The pheromone parameter is updated.

This technique has a big inconvenient: its flexibility. If a new shortest path is found, it may not be chosen by the ants since there are some previous paths with a higher pheromone quantity that will lead the ants to choose them. [11]

Like the other Metaheuristics, this one will be repeated until the stop criterion is reached.

3.3.4 Tabu Search

Tabu Search was presented by Glover in 1986. The big advantage of this method is the possibility of learn the past lessons like the human memory. [11]

Associated to this technique there are a lot of critical characteristics that can vary from problem to problem [7]:

- Search Space
- Initial Solution
- Attribute set of a solution
- Neighbourhood
- Tabu Duration
- Diversification
- Stopping Criterion

A more detailed explanation of this Metaheuristic is presented on chapter 4. This approach is the one chosen to solve the proposed problem in this Master Thesis.

3.4 Summary

The chapter called Methods of Optimization provides a large view of the three major types of optimization methods: Exact Methods, Heuristics and Metaheuristics.

In the section referred to the Exact Methods are explained two of these Methods: Branch and Bound and Branch and Cut. Although these methods are capable of providing the optimal solution to some problems, they demand a lot of computational time and for some problems they can't find the optimal solution due to this aspect. Last subsection of this section contains a brief introduction to Dijkstra Algorithm. This algorithm provides the minimum path between two distinct nodes.

The Heuristics may or may not provide the optimal solution to a given problem so when applying a heuristic we need to verify two different parameters to examine the quality of the method: quality of the solution and computer time needed to find the final solution. The heuristic's purpose is to find a solution near to the optimal one in the shortest possible time. The big three groups studied in this chapter are: Constructive Heuristics, Improvement Heuristics and Compound Heuristics.

The aim of the Constructive Heuristic is to add points to the final solution until all of them are present in it. The Improvement Heuristics aims obtaining a better solution having as starting point a first solution that can be built through the application of a Constructive Heuristic. Finally, Compound Heuristics is composed by all the methods that enclose more than one phase to obtain a final solution to a given problem.

Finally, the last section contains a brief introduction to a set of very important methods called Metaheuristics. These methods are used to solve a lot of different combinatorial optimization problems. The biggest disadvantages associated with these methods are the parameter adjustment and the computational time demanded to find a solution. The Metaheuristics presented in this chapter are: Simulated Annealing, Genetics Algorithms, Ant Colony Algorithms and Tabu Search. These were chosen to provide a large spectrum analogies.

After reading this chapter, reader should be able to understand the main aspects associated to the Optimization Methods here presented and its importance in the optimization problems field.

Chapter 4

Optimization Models and Methodology

As mentioned in Chapter 2 the problem is related to MCARP (Mixed Capacitated Arc Routing Problems) since it's composed by a mixed graph and has capacity constraints associated to it. Each arc/edge may have different costs associated: deadheading cost and, if it's required, a service cost

This chapter is divided into three different sections. The first one contains the concepts used in the further chapters for the problem in analysis.

The second section contains the notation used in the literature and three possible Optimization Models to deal with the problem. The first model presented is the only one capable of providing a feasible solution.

Finally, in the last section, it's presented a detailed explanation of the different phases that compose the methodology followed in this work.

4.1 Problem Characteristics

Before the presentation of the notation and the relevant optimization models, it's presented the different concepts used in the present chapter and in the work developed by [18].

In the studied problem the street network is described by a mixed graph composed by:

- Edges that symbolize the two way streets where zig-zag collection is allowed;
- Arcs that represent the one way streets or large two way streets where zig-zag collection is not allowed;
- Nodes that represent the street crossings or dead-end streets.

In this graph exists a special node, named depot, that is the starting and ending point for every trips and is the local where the vehicles empty the collected refuse.

A vehicle trip is a circuit that has its beginning and its ending in the depot. While the vehicle doesn't return to the depot it is responsible for servicing the streets without exceeding its maximum capacity.

The streets that need to be served are designated by required links or tasks and the ones that don't have a demand associated and are traversed only to guarantee the connectivity of the trips are known as deadheading streets.

To simplify the problem it's assumed that each available vehicle performs only one trip.

4.2 Optimization Models

This section shows three different models that can be used to deal with the problem studied in this Master Thesis.

The first one, OM1 (Optimization Model 1), is the only one that it's capable to provide a feasible solution to this kind of problems. Nevertheless, this is not the model used since it needs a high computational effort and it's not guaranteed that it is able to obtain the optimal solution to the respective problem.

The last two models, OM2 and OM3, provide an infeasible solution that doesn't take the vehicle capacity constraint into consideration.

Each OM is composed by a set of constraints and an objective function. The main difference between these three models lies in the constraints attached to each one since the objective function keeps the same for all of them.

Before the presentation of these optimization models, we can find the notation used by [18]:

- $\Gamma = (N, A' \bigcup E)$ that represents the mixed graph where $A_R \subseteq A'$ is the set of required arcs and $E_R \subseteq E'$ the set of required edges; N represents the nodes of the problem.
- $0 \in N$ is the depot node.
- $G = (N, A)$ is a directed graph where each edge from E is replaced by two opposite arcs.
- $R \subseteq A$ represents the set of required arcs in G ($|R| = |A_R| + 2|E_R|$)
- P represents the maximum number of trips.
- W is the capacity of each vehicle.
- λ represents the disposal cost, activated when a vehicle is emptied at the depot.
- d_{ij} is the deadheading cost of the arc (i, j).
- c_{ij} is the service cost related with servicing arc (i, j).
- q_{ij} is the demand of arc $(i, j) \in A$.
- Q_T is the total demand given by: $Q_T = \sum_{(i,j) \in A_R \bigcup E_R} q_{ij}$

4.2.1 Optimization Model 1 (OM1) :

The OM1 is composed by a set of constraints and an objective function that provides a feasible final solution. This solution is composed by a certain number of routes, the same as the number of available vehicles considered for the problem under analysis.

As all the OM presented in this section, the OM1's purpose is to minimize the value of the objective function. This includes all the costs associated to the existing arcs/edges of the problem: deadheading cost and service cost if the arc is required and when applied an additional cost, named disposal cost.

Below is presented the OM1 drawn from [18]:

Objective Function:

$$min \sum_{p=1}^{P} [\sum_{(i,j)\in R} c_{ij} x_{ij}^p + \sum_{(i,j)\in A} d_{ij} y_{ij}^p + \lambda \sum_{(i,0)\in A} y_{i0}^p + \lambda \sum_{(i,0)\in R} x_{i0}^p] \tag{4.1}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij}^p + \sum_{j:(i,j)\in R} x_{ij}^p = \sum_{j:(j,i)\in A} y_{ji}^p + \sum_{j:(j,i)\in R} x_{ji}^p \qquad i = 0, 1, ..., n; p = 1, ..., P \tag{4.2}$$

$$\sum_{p=1}^{P} x_{ij}^p = 1 \qquad \forall (i, j) \in A_R \tag{4.3}$$

$$\sum_{p=1}^{P} (x_{ij}^p + x_{ji}^p) = 1 \qquad \forall (i, j) \in E_R \tag{4.4}$$

$$\sum_{j:(0,j)\in A} y_{0j}^p + \sum_{j:(0,j)\in R} x_{0j}^p \leq 1, \qquad p = 1, ..., P \tag{4.5}$$

$$\sum_{j:(j,i)\in A} f_{ji}^p - \sum_{j:(i,j)\in A} f_{ij}^p = \sum_{j:(j,i)\in R} q_{ji} x_{ji}^p, \qquad i = 1, ..., n; \quad p = 1, ..., P \tag{4.6}$$

$$\sum_{j:(0,j)\in A} f_{0j}^p = \sum_{(i,j)\in R} q_{ij} x_{ij}^p, \qquad p = 1, ..., P \tag{4.7}$$

$$\sum_{i:(i,0)\in A} f_{i0}^p = \sum_{i:(i,0)\in R} q_{i0} x_{i0}^p, \qquad p = 1, ..., P \tag{4.8}$$

$$f_{ij}^p \leq W(y_{ij}^p + x_{ij}^p) \quad \forall (i, j) \in A, \qquad p = 1, ..., P \tag{4.9}$$

$$x_{ij}^p \in \{0, 1\} \quad \forall (i, j) \in R, \qquad p = 1, ..., P \tag{4.10}$$

$$f_{ij}^p \geq 0 \quad \forall (i, j) \in A, \qquad p = 1, ..., P \tag{4.11}$$

$$y_{ij}^{p} \geq 0 \qquad integer \quad \forall (i,j) \in A, \qquad p = 1,...,P \tag{4.12}$$

The first equation, 4.1, symbolizes the objective function that is calculated by the sum of all the costs: deadheading cost, service cost and disposal cost.

The equation 4.2 guarantees the continuity of the trips on each vertex.

The constraint 4.3 is responsible for ensuring that all the required arcs are served by the vehicles and the constraint 4.4 has the same purpose but relatively to the required edges.

The constraint 4.5 guarantees the correct accounting of the disposal cost in the objective function.

Equations 4.6, 4.7 and 4.8 (flow conservation constraints) with equation 4.9 (linking constraint) force the connectivity of the trips.

The constraint 4.6 guarantees that if the arc (j,i) is served by vehicle p, the demand q_{ji} will be absorbed by the flow f_{ji}.

Finally, the last constraint will impose upper bounds on the flow variables needed to guarantee that the capacity constraint associated with the vehicle is followed.

Even though this model is the correct and the only one presented in this section capable of providing a feasible solution, it is not able to provide a solution in an acceptable computational time for some problems. The other disadvantage associated with this model is the uncertainly of finding the optimal solution due to the time needed to reach it. [18] These different reasons leads the author to not use this OM to deal with the proposed problem.

4.2.2 Optimization Model 2 (OM2) :

This model is used in the work developed by [18] and, as the previous one here presented, aims for the minimization of the value of the objective function.

The solution provided by this OM is composed by one route travelled by one vehicle with no limitation on the total amount of USW (Urban Solid Waste) collected.

Objective Function:

$$min \sum_{(i,j)\in R} c_{ij}x_{ij} + \sum_{(i,j)\in A} d_{ij}y_{ij} + \lambda \sum_{(i,0)\in A} y_{i0} + \lambda \sum_{(i,0)\in R} x_{i0} \tag{4.13}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij} + \sum_{j:(i,j)\in R} x_{ij} = \sum_{j:(j,i)\in A} y_{ji} + \sum_{j:(j,i)\in R} x_{ji}, \qquad i = 0,1,...,n \tag{4.14}$$

$$x_{ij} = 1 \qquad \forall (i,j) \in A_R \tag{4.15}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall (i,j) \in E_R \tag{4.16}$$

$$\sum_{j:(0,j)\in A} y_{0j} + \sum_{j:(0,j)\in R} x_{0j} \leq P \tag{4.17}$$

$$\sum_{j:(j,i)\in A} f_{ji} - \sum_{j:(i,j)\in A} f_{ij} = \sum_{j:(j,i)\in R} q_{ji} x_{ji}, \qquad i = 1,\ldots,n \tag{4.18}$$

$$\sum_{j:(0,j)\in A} f_{0j} = Q_T \tag{4.19}$$

$$\sum_{i:(i,0)\in A} f_{i0} = \sum_{i:(i,0)\in R} q_{i0} x_{i0} \tag{4.20}$$

$$f_{ij} \leq W(y_{ij} + x_{ij}) \qquad \forall (i,j) \in A \tag{4.21}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in R \tag{4.22}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{4.23}$$

$$y_{ij} \geq 0 \quad integer \qquad \forall (i,j) \in A \tag{4.24}$$

There are two main differences between this OM and the one presented before. These differences lie on two distinct constraints:

- Constraint 4.19 that ensures that the total amount of USW is collected by the only vehicle considered by the model.

- Constraint 4.17 is responsible for guaranteeing that the solution contains the same entries/exits from the depot as the number of available vehicles.

As will be explained in the section concerning the methodology, the model used aims the construction of the first solution that is used by the other phases of the approach followed. Due to this fact it's irrelevant the use of a solution comprising a certain number of entrances/exits of the depot since it's just required to create a solution with one route. That's the reason why this model is not the one applied to deal with the problem.

The model adopted to solve the proposed problem is presented and explained in the next section.

4.2.3 Optimization Model 3 (OM3) :

The solution obtained through the application of this model is similar to the solution given by the previous model. The main difference lies on the way that this route is built. Since, as aforementioned, the solution obtained by this model doesn't include the same number of entries/exits from the depot as the number of available vehicles. It's just composed by one entry and one exit from this special node.

This model was constructed taking as starting point the two models exposed before drawn from work developed by [18]. Next, is presented the OM3:

Objective Function:

$$min \sum_{(i,j)\in R} c_{ij}x_{ij} + \sum_{(i,j)\in A} d_{ij}y_{ij} + \lambda \sum_{(i,0)\in A} y_{i0} + \lambda \sum_{(i,0)\in R} x_{i0} \tag{4.25}$$

Subject to:

$$\sum_{j:(i,j)\in A} y_{ij} + \sum_{j:(i,j)\in R} x_{ij} = \sum_{j:(j,i)\in A} y_{ji} + \sum_{j:(j,i)\in R} x_{ji} \qquad i = 0,1,\ldots,n; \tag{4.26}$$

$$x_{ij} = 1 \qquad \forall (i,j) \in A_R; \tag{4.27}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall (i,j) \in E_R; \tag{4.28}$$

$$\sum_{j:(0,j)\in A} y_{0j} + \sum_{j:(0,j)\in R} x_{0j} = 1; \tag{4.29}$$

$$\sum_{j:(j,i)\in A} f_{ji} - \sum_{j:(i,j)\in A} f_{ij} = \sum_{j:(j,i)\in R} q_{ji}x_{ji}, \qquad i = 1,\ldots,n; \tag{4.30}$$

$$\sum_{i:(i,0)\in A} f_{i0} = \sum_{i:(i,0)\in R} q_{i0}x_{i0}; \tag{4.31}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in R; \tag{4.32}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in A; \tag{4.33}$$

$$y_{ij} \geq 0 \quad integer \qquad \forall (i,j) \in A \tag{4.34}$$

This model differs from the previous one due to two main differences:

- Two constraints of the previous model are not considered in this OM – Constraint 4.19 and 4.21;

- Constraint 4.29 differs from the corresponding constraint in OM2 (constraint 4.17) since it's responsible for eliminating the number of entries/exits from the depot considered in the previous OM.

In this Master Thesis, in order to obtain the final solution to the instances that composes the proposed problem, it's used a Methodology applied in three different phases. In the first phase of this approach it's used an Optimization Model to obtain the first solution to the problem. The OM chosen to apply during this phase is OM3.

4.3 Adopted Methodology

To find the final solution of the problem under study it was developed a methodology composed by three phases, as already mentioned in Chapter 1. This section of the present chapter contains the crucial explanations to provide a full comprehension about the methods used by the author.

The methodology developed is composed by three distinct phases. Each one provides a solution that is used as an input by the next phase. Each phase has a name associated:

- Phase 1 : Mega-Route

- Phase 2 : Intermediate Solution

- Phase 3 : Final Solution

These names were adopted to provide a better understanding when, in a certain phase, is referred a solution obtained by a different phase.

A problem-example will be presented in order to better exemplify each phase.

This problem-example is a mixed graph with:

- 5 nodes;

- 3 required arcs where each one has a demand of 2;

- 2 available vehicles with a capacity of 4.

The remaining characteristics of this problem are shown in Table 4.1.

Table 4.1: Characteristics of the problem-example

Arcos	Deadheading Cost	Service Cost	Demand
(1 - 2)	2	2	0
(2 - 1)	2	2	0
(2 - 3)	2	2	2
(2- 4)	2	2	2
(2 - 5)	2	2	2
(3 - 2)	2	2	0
(4 - 2)	2	2	0
(5 - 2)	2	2	0

The following sections contain the explanations of the different phases of the methodology.

4.3.1 Phase 1: Mega-Route

The name given to the solution obtained by this phase is Mega-Route due its characteristics. This solution, as aforementioned, is composed by one route travelled by one vehicle.

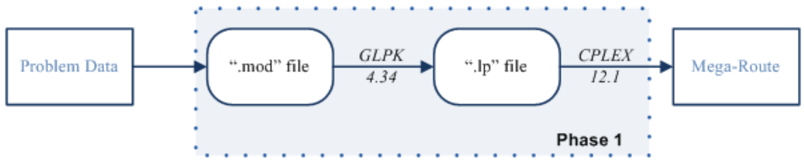


Figure 4.1: Methodology - Phase 1

The first phase of the methodology can be divided into two different stages as illustrated in figure 4.1:

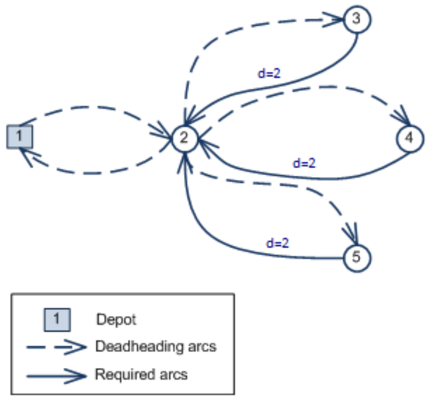
- Stage 1: Use of an optimization program to convert a Model file (mod format) into a Linear Programming file (lp format):
  - It's written a Model file containing the OM3 described in the previously in section 4.2.
  - Use of GLPK 4.34 [25], an optimization program, aiming the conversion of the Model file into a Linear Programming file.
- Second Stage: Use of another optimization program to obtain the Mega-Route solution.
  - Uses the LP file obtained by GLPK to acquire Mega-Route through the application of another optimization program, called CPLEX 12.1 [31] .

Concerning the CPLEX, it is not able to obtain solutions to Model files due to it's particularities, and that is the reason that leads to the transformation performed in the First Stage of the approach followed.

The file is initially written in the Model format since it's much less complex to write than LP files.

The solution obtained by these different stages explained before is used as an input by the next phase of the methodology.

Next is shown the solution obtained in this phase to the problem-example presented in the beginning of this section.



Route	Demand
(1-2)	0
(2-3)	0
(3-2)	2
(2-4)	0
(4-2)	2
(2-5)	0
(5-2)	2
(2-1)	0
Total Demand	6

Figure 4.2: Solution obtained to problem-example by the first phase of the methodology

As observed in Figure 4.2 the solution obtained is composed by one route that contains all the required arcs of the problem. The table contained by the same figure shows the arcs that compose the route travelled by the vehicle, the demand collected in each arc and the total demand collected.

The value of the objective function is 16, and it's calculated by:

$$fo_{Phase1} = d_{12} + d_{23} + c_{32} + d_{24} + c_{42} + d_{25} + c_{52} + d_{21} = 16$$

4.3.2 Phase 2: Intermediate Solution

The solution obtained previously – Mega-Route – is saved into a text file and it's used as an input by this phase.

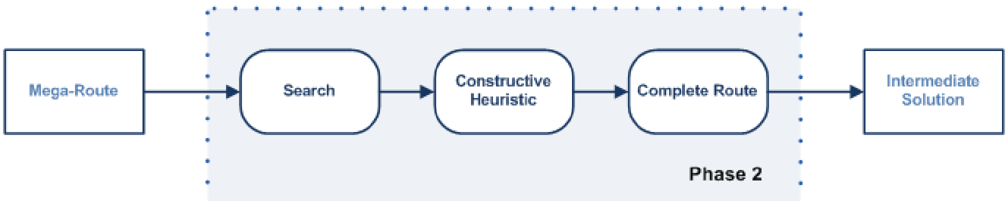


Figure 4.3: Methodology - Phase 2

As observed in Figure 4.3 this phase can be characterized as a composition of three different functions.

The Figure 4.4 shows the algorithm applied in this phase and as observed it contains all the functions mentioned previously. These functions are performed as many times as a certain number of iterations defined by the author.

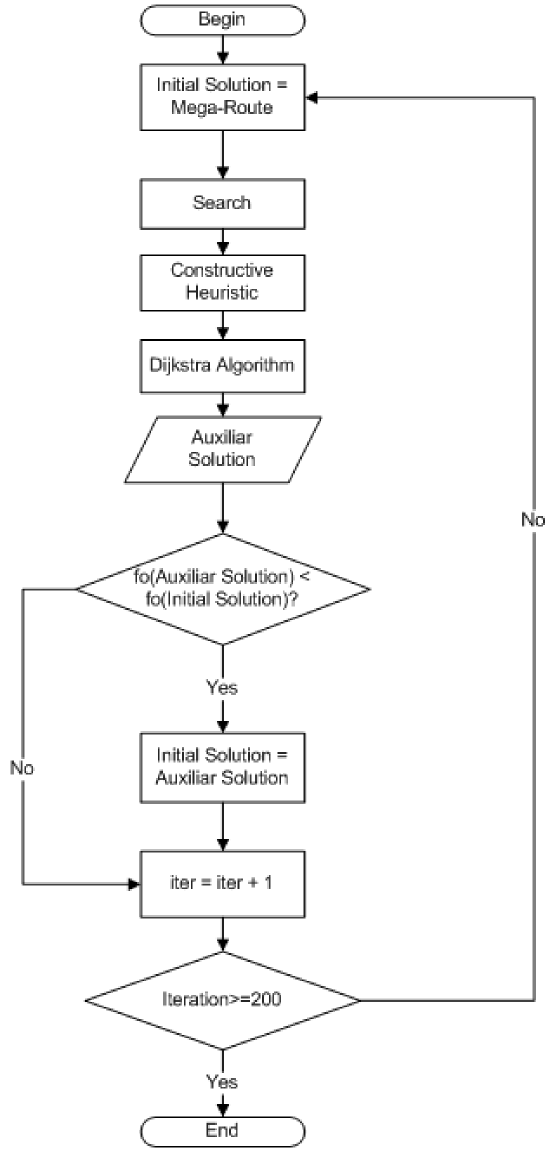


Figure 4.4: Algorithm used by Phase2

Further on follows a detailed explanation of the different functions represented in the previous two figures, Figure 4.3 and 4.4 :

- Search;
- Constructive Heuristic;

- Complete Route.

The solution provided by this phase is the first feasible solution of the methodology since the different routes that compose it, are constructed taking into consideration the capacity restriction associated with the different available vehicles.

As observed in Figure 4.4 these different functions are applied as many times as the number of iterations. When this number is reached the algorithm used in this phase should stop and the final solution of this phase, called Intermediate Solution, is obtained.

Next is presented the explanation about the three distinct functions aforementioned - Search, Constructive Heuristic and Complete Route.

Search

The main purpose of Search function here applied is to randomly change the order of the Mega-Route solution obtained by the previous phase.

Before applying this function it's necessary to read the Mega-Route solution from the respective file. Without this information the final solution can't be built.

The Search function changes the order by which the arcs of the previous solution are travelled.

This function is used to search different possible solutions to the same problem. Some of them are worse than the first one and others are better.

The next figure (Figure 4.5) presents all the steps performed in this function:

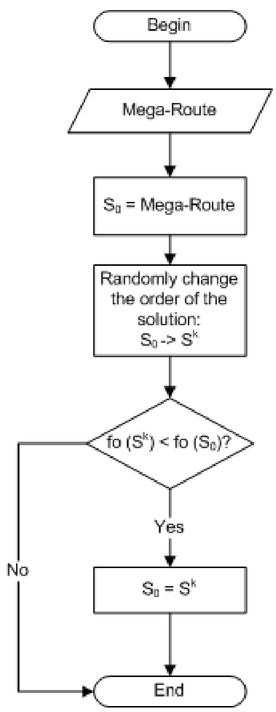


Figure 4.5: Algorithm used by Search function

Like the Local Search Heuristic presented on Chapter 3 this function aims to obtain a better solution searching on the neighbourhood of the first solution obtained (Mega-Route).

Constructive Heuristic

Having as an input the solution obtained by the previous function, this Heuristic aims to divide required arcs by the available vehicles.

The algorithm followed by this Heuristic is explained below:

- $Step1$: Set the first arc of the previous solution as the current arc and the first available vehicle as the current vehicle.

- $Step2$: If there aren't any remaining arcs to add to the final solution go to $Step4$.

- $Step3$: If the current arc can be added to the final solution without exceeding the vehicle capacity go to $Step3.1$. Else go to $Step3.2$.

  - $Step3.1$: Add current arc to the final solution. Actualize the total demand collected by the current vehicle. Go to $Step2$.

  - $Step3.2$: Select the next vehicle as the current one. If there's no other available vehicle go to $Step4$. Else go to $Step2$.

- $Step4$: Final Solution found.

The solution acquired by the algorithm explained previously isn't a valid solution since it's composed by some infeasible paths. A path is considered infeasible when the final node of the last visited arc doesn't correspond to the initial node of the next arc to be visited. To face this issue, it's implemented another function called Complete Route that is explained below.

Complete Route

To complete the routes obtained previously, this function uses Dijkstra Algorithm.

As referred before, in Chapter 3, this algorithm obtains the minimum path between two distinct nodes. In the ARP (Arc Routing Problem) this minimum path is obtained taking into account the cost of transverse a set of existing arcs to go from source to the destine node. The value obtained to the total cost of transverse the possible paths from a node to another is calculated by the sum of deadheading costs of the arcs that need to be transverse.

Below is explained the algorithm used by Dijkstra algorithm:

- $Step1$: Initial Node marked with definitive label = 0

- $Step2$: Other nodes marked with a provisory label = ∞

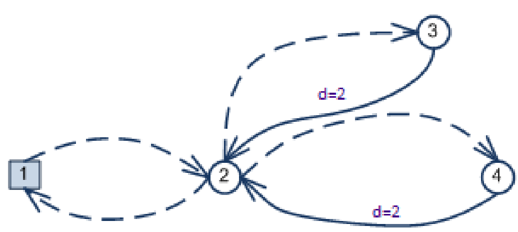
- $Step3$: Being k the node that has received the last definitive label:

- *Step*3.1 : Calculate the direct distance between k and the other nodes $i - d_{ik}$
- *Step*3.2 : Calculate the sum of the definitive label of k and d_{ik}.

- *Step*4: Select the minimum value between d_{ik} and the value of the last provisory label of i. Take the minimum value as the new value to the provisory label of node i.
- *Step*5: Mark as definitive the label with the minimum value obtained in *Step*4.
- *Step*6: If the label marked is the label of the destine node – go to *Step*7. Else go to *Step*3.
- *Step*7: Minimum Path found.

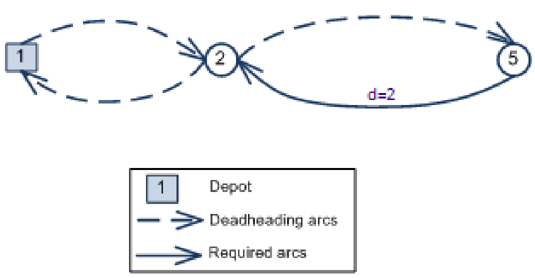
The solution acquired in the end of this phase for the problem-example presented in the beginning of this section is presented in Figure 4.3 :

Route 1



Route 1		Demand
(1 - 2)	- ->	0
(2 - 3)	- ->	0
(3 - 2)	→	2
(2 - 4)	- ->	0
(4 - 2)	→	2
(2 - 1)	- ->	0
Total Demand		4

Route 2



Route 2		Demand
(1 - 2)	- ->	0
(2 - 5)	- ->	0
(5 - 2)	→	2
(2 - 1)	- ->	0
Total Demand		2

Figure 4.6: Solution obtained for the problem-example by the second phase of the methodology

As observed previously this solution has the following characteristics:

- Contains two routes travelled by two different vehicles;
- The vehicle capacity is not exceeded in any of the two routes;
- All the required arcs are served;
- The total Demand – 6 – is collected.

To this example the value obtained for the objective function is:

$$fo_{Phase2} = d^1_{12} + d^1_{23} + c^1_{32} + d^1_{24} + c^1_{42} + d^1_{21} + d^2_{12} + d^2_{25} + c^2_{52} + d^2_{21} = 20$$

This is a very basic problem since it's composed by few nodes and arcs, so the solution obtained in this phase is the optimal solution of the respective problem. Even though, next phase will be applied to this Intermediate Solution.

The value obtained by the previous phase of this approach was lower than the optimal solution verified to this problem. This happens due to the characteristics of the solution returned by the previous phase.

4.3.3 Phase 3: Final Solution

The final phase of the methodology here presented is characterized by an application of a Metaheuristic (Figure 4.7) used to increase the quality of the solution already found by the previous phases.

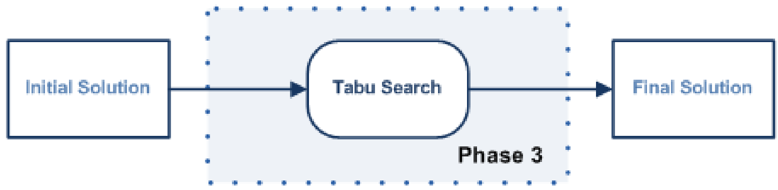


Figure 4.7: Methodology - Phase 3

In Figure 4.8 is presented the algorithm used to obtain the Final Solution.

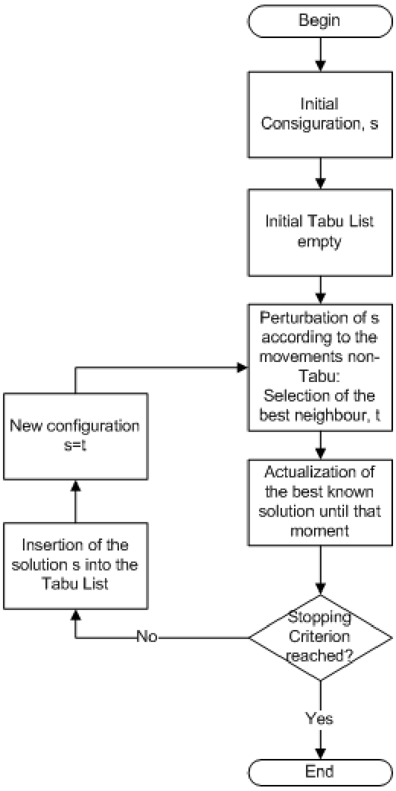


Figure 4.8: Algorithm used by Tabu Search

Like referred in Chapter 3 there are some aspects that are crucial to guarantee that the Tabu Search works properly.

Below are exposed and explained a set of characteristics considered fundamental to the way that this method is applied.

1. Tabu Duration

   Tabu duration is characterized by the size of Tabu List. This list contains the last solutions visited by the method.

   If the size of this list is too short, the method may be returning to the same local optimum, however if this size is too long this will need an excessive computational time to decide if a movement is Tabu or not. [33]

   The size chosen by the author to this list is 7 since this number is not considered too short or too long. Many works already developed by other authors use a Tabu List with this size obtaining satisfactory results to the problem under study. [33]

2. Aspiration Criterion

   This characteristic is defined to enable the acceptance of a solution even if it's present in the Tabu List.

In the method here applied a solution is accepted even if it's present on the Tabu List, if its objective function is smaller than the value of the aspiration function.

3. Neighbourhood Search

This search has the aim of finding solutions in the neighbourhood of the one obtained previously. To achieve this it's used an exchange heuristic that is responsible to swap two required arcs between two different routes.

The three basic steps used to realize these changes on the solution are presented bellow.

- $Step1$: Randomly select two different routes of the solution – $Route1$ and $Route2$;
- $Step2$: Randomly select two distinct positions of $Route1$ and $Route2$;
- $Step3$: If this change is possible - if the required arcs from $Route1$ can be inserted on $Route2$ and the opposite without exceeding the vehicle capacity - the swap is realized. Otherwise go to $Step3.1$
  - $Step3.1$: If these changes were tried more than ten times without being executed go to $Step1$. Else go to $Step2$.

This heuristic is applied in all the iterations realized by the method since its objective is to produce some changes in the solution.

4. Diversification

The main purpose of this characteristic is to produce a change in the solution, searching on its neighbourhood.

Crossover is an operator that aims the creation of a new solution through the merger of two initial solutions (parents).

To reach this goal it's used a Crossover Operator based on work developed by [29]:

- $Step1$: Two different solutions – S_1 e S_2 ;
- $Step2$: Select crossover point, u– This point is randomly selected between two different pre-determined positions: $0.4N$ and $0.6N$, where N represents the number of nodes of the instance to test.
- $Step3$: New solution constructed:
  - $Step3.1$: First u required arcs of S_1 is kept in the same order - S'
  - $Step3.2$: When an arc from S_2 isn't present in S_1, it's introduced in S' after other remaining arcs, keeping the order followed in S_2.

This operator is applied in 5000 to 5000 iterations.

5. Stopping Criterion

   This aspect is responsible for defining when the method should stop. The criterion used on this methodology is if isn't verified any improvement in the solution at the end of 100000 iterations, Metaheuristic stops and the final solution is returned.

After the application of the different phases of the methodology under the problem-example the solution obtained is the same as the one obtained on the end of the second phase. This happens due to the simplicity of this problem.

4.4 Summary

After reading this chapter, the reader should be able to understand all the principles associated to the approach used to deal with the problem under concern.

The first section contains three Optimizations Models where just the first one is capable of providing a feasible solution, but will demand a lot of computational effort and it's not guaranteed that the optimal solution is found.

The other two models produce infeasible solutions since these are composed by just one route and don't take into consideration the capacity restriction. The difference between these two models lay on the way that this route is built. In the first model (OM2) the construction of the final solution includes the entries and exits of the depot and its number is equal to the number of available vehicles. The last model presented, OM3, doesn't comprise these entries/exits from the depot.

The second section of this chapter comprises the explanation of the methodology used to solve the problem. This methodology was applied in three distinct phases. The first one uses optimization programs with the intent of obtaining an infeasible solution to the problem. This is obtained taking the OM used as a starting point.

The second phase of the methodology is applied with the aim of obtaining the first feasible solution, called Intermediate Solution, to the respective problem. This phase comprises three distinct functions: Search, a Constructive Heuristic and a function called Complete Route. This phase uses the solution obtained by the previous phase as an input.

Finally, the last phase of the approach includes the application of a Metaheuristic. This method aims to increase the quality of the solution already given by Phase 2. The Metaheuristic used is the Tabu Search. This method can be characterized by a certain number of aspects like: Tabu Duration, Aspiration Criterion, Neighbourhood Search, Diversification and Stopping Criterion. These different aspects will influence directly the quality of the solution acquired.

Chapter 5

Computational Results

This chapter intends to show the computational results obtained during the realization of this Master Thesis.

The problem under study is composed by a several number of instances. Regardless, the basic principles of all of them are the same, their characteristics will be different from instance to instance. The different characteristics of the problem (capacity, number of available vehicles, demands on each street, service and deadheading costs associated with distinct streets and dump costs) are known for every tested instance.

This problem is composed by two distinct sets of instances. The first set, *mval* instances, contains the ones that have a fewer number of nodes (24-50) and links (43-138) which are all required. The second set, the *lpr* instances, contains a higher number of nodes (28-401) and a higher number of links (50-1056) but just a subset of these are required.

The present chapter is divided into two main sections. Each one presents the characteristics of different instances and then the results attained in each phase of the methodology explained in the previous chapter, Chapter 4. The results obtained through the application of this approach are compared with the best known value given by the work developed by [18] to evaluate the quality of the approach followed in this work.

The different tests were performed on an Intel(R) Core(TM) i5 CPU 2.40GHz with 4.00GB RAM.

5.1 *mval* Instances

This set of instances is characterized for different aspects exposed below:

- All the vehicles begin and end their trips on a special node, called depot.
- When servicing an arc/edge, this will be served just one time by the assigned vehicle.
- The deadheading cost and the service cost of each arc/edge will be different from instance to instance.

- The disposal cost isn't charged on these instances.
- The objective function of the problem is given by the sum of two different costs: deadheading cost and service cost since the disposal cost is no charged in this set of instances.
- The number of required edges is always greater than the number of required arcs.

After the description of the main characteristics of these instances, the main values of the different parameters are presented in Table 5.1.

Table 5.1: Characteristics of the *mval* instances

File	$\|N\|$	$\|A \cup E\|$	$\|A_r\|$	$\|E_r\|$
mval1A	24	55	20	35
mval1B	24	51	13	38
mval1C	24	53	17	36
mval2A	24	44	16	28
mval2B	24	52	12	40
mval2C	24	49	14	35
mval3A	24	48	15	33
mval3B	24	45	16	29
mval3C	24	43	18	25
mval4A	41	95	26	69
mval4B	41	102	19	83
mval4C	41	103	21	82
mval4D	41	104	21	83
mval5A	34	96	22	74
mval5B	34	91	35	56
mval5C	34	98	17	81
mval5D	34	92	29	63
mval6A	31	69	22	47
mval6B	31	66	22	44
mval6C	31	68	23	45
mval7A	40	86	36	50
mval7B	40	91	25	66
mval7C	40	90	28	62
mval8A	30	96	20	76
mval8B	30	91	27	64
mval8C	30	83	28	55
mval9A	50	132	32	100
mval9B	50	120	44	76
mval9C	50	125	42	83
mval9D	50	131	38	93
mval10A	50	138	32	106
mval10B	50	134	33	101
mval10C	50	136	36	100
mval10D	50	129	42	87

Column 2 of the Table 5.1 contains the number of the nodes of each instance. Column 3 shows the set of arcs and edges of the respective instance. The last two columns, 4 and 5, show the set of required arcs and required edges, respectively.

The results obtained to these instances are divided into three subsections, each one related with the corresponding phase of the methodology.

5.1.1 Phase1 - Mega-Route

Table 5.2 shows the results obtained to this set of instances tested.

Table 5.2: Phase 1 - Results obtained to *mval* Instances

Phase 1 - Mega-Route		
File	**Nodes**	**Objective Function**
mval1A	24	230
mval1B	24	261
mval1C	24	255
mval2A	24	324
mval2B	24	351
mval2C	24	313
mval3A	24	113
mval3B	24	128
mval3C	24	88
mval4A	41	566
mval4B	41	616
mval4C	41	609
mval4D	41	616
mval5A	34	597
mval5B	34	581
mval5C	34	670
mval5D	34	705
mval6A	31	326
mval6B	31	313
mval6C	31	303
mval7A	40	364
mval7B	40	412
mval7C	40	393
mval8A	30	581
mval8B	30	531
mval8C	30	527
mval9A	50	458
mval9B	50	453
mval9C	50	426
mval9D	50	478
mval10A	50	630
mval10B	50	653
mval10C	50	615
mval10D	50	562

The results presented above are divided into three columns that contain information about the number of the nodes (Column 2) and the value obtained in this phase to the objective function of the respective instance (Column 3).

These results were obtained through the use of some functions belonging to the C++ libraries from the CPLEX software. These existing functions were incorporated with the C++ code written using Visual Studio 2008 C++.

5.1.2 Phase2 - Intermediate Solution

The computational results obtained to this phase are shown in Table 5.3.

Table 5.3: Phase 2 - Results obtained to *mval* Instances

Phase 2 - Intermediate Solution			
File	**Objective Value**	**Total Demand**	**Number of Vehicles**
mval1A	272	358	2
mval1B	335	358	3
mval1C	413	328	8
mval2A	362	310	2
mval2B	427	310	3
mval2C	684	310	8
mval3A	121	137	2
mval3B	167	137	3
mval3C	196	137	7
mval4A	648	627	3
mval4B	788	627	4
mval4C	811	627	5
mval4D	1039	627	9
mval5A	689	614	3
mval5B	688	614	4
mval5C	815	614	5
mval5D	1118	614	9
mval6A	368	451	3
mval6B	368	451	4
mval6C	475	451	10
mval7A	434	559	3
mval7B	493	559	4
mval7C	603	559	9
mval8A	663	566	3
mval8B	633	566	4
mval8C	872	566	9
mval9A	556	654	3
mval9B	528	654	4
mval9C	545	654	5
mval9D	746	654	10
mval10A	705	704	3
mval10B	782	704	4
mval10C	766	704	5
mval10D	888	704	10

Table 5.3 contains information about the available vehicles (column 4), total demand collected (Column 3) and the value obtained by this phase for the objective function of the respective instance (Column 2).

5.1.3 Phase3 - Final Solution

Results obtained in this phase are exposed in Table 5.4.

Table 5.4: Phase 3 - Results obtained to *mval* Instances

Phase 3 - Final Solution				
File	**Best known value**	**Objective Value**	**Time(sec)**	**Gap(%)**
mval1A	230*	245	272	6,52%
mval1B	261*	277	233	6,13%
mval1C	309*	357	734	15,53%
mval2A	324*	350	420	8,02%
mval2B	395*	413	193	4,56%
mval2C	521*	602	700	15,55%
mval3A	115*	119	352	3,48%
mval3B	142*	151	750	6,34%
mval3C	166*	192	669	15,66%
mval4A	580*	640	660	10,34%
mval4B	650*	718	677	10,46%
mval4C	630*	762	851	20,95%
mval4D	746	949	1263	27,21%
mval5A	597*	659	809	10,39%
mval5B	613*	685	502	11,75%
mval5C	697*	786	875	12,77%
mval5D	719	928	2124	29,07%
mval6A	326*	337	355	3,37%
mval6B	317*	356	5206	12,30%
mval6C	365	457	666	25,21%
mval7A	364*	407	1079	11,81%
mval7B	412*	483	774	17,23%
mval7C	424	546	1117	28,77%
mval8A	581*	658	2147	13,25%
mval8B	531*	601	524	13,18%
mval8C	617	795	1368	28,85%
mval9A	458*	532	1020	16,16%
mval9B	453*	524	764	15,67%
mval9C	428	524	1262	22,43%
mval9D	514	696	7709	35,41%
mval10A	634*	694	935	9,46%
mval10B	661*	730	1170	10,44%
mval10C	623*	727	1405	16,69%
mval10D	643	829	2248	28,93%
Average			1230,38	15,41%
Maximum			7709	35,41%
Minimum			193	3,37%

In the above table are shown the final results for the instances under test. The second column shows the best known results obtained to the respective instance and if it's the optimal value is marked with an (*). These results were demonstrated by [18].

Column 3 contains the values achieved by the approach used in this Master Thesis. Column 4 has the time in seconds and finally, the last column contains the difference in percentage between the best known result and the value obtained by the approach followed in this work.

Results Discussion

Bellow is presented the discussion of the results attained by the new approach developed during this Master Thesis. These results were compared with the ones obtained by [18] as already mentioned.

The first phase of the methodology provides lower values for the objective function than the best known value for the same instance. This happens because the solution retrieved by this phase is an infeasible one composed by just one route travelled by one vehicle.

The solutions obtained in the second phase have values much higher than the best known values of the respective instances and that's the reason why other phase is applied. Since this solution is the one used as an input in the next phase, if it has a very high value for the objective function the Metaheuristic applied in the next phase may not be able to find a solution near to the optimal one.

The difference in percentage, called gap, between the best known result and the result achieved by this approach are depicted in the last column of the Table 5.4 and is calculated by:

$$[(BestKnownValue - ObjectiveValue) \div BestKnownValue] \times 100$$

The approach applied to obtain a solution to this set of instances doesn't provide the optimal value to none of them, as already observed in Table 5.4. As previously explained, Metaheuristics are not always able to provide the optimal solution to a given problem since it's an approximative method.

The worst scenarios were obtained for the more complex instances with a higher number of available vehicles - *mval4D, mval5D, mval6C, mval7C, mval8C, mval9D* and *mval10D* - as shown in Table 5.4. The results attained to these instances are between 15% and 36%.

The gap values verified to the instances *mval1A, mval1B, mval2A, mval2B, mval3A, mval3B* and *mval6A* are no greater than 10%. And the best value obtained is for the instance *mval6A*, with a gap near to 3.4%. In general, to the instances with a least number of vehicles are obtained good final results.

The average gap obtained to these different instances is near to the 15%, with a maximum value of 35.41% to the instance called *mval9D* that is composed by 50 nodes and 10 available vehicles.

Finally, the average CPU time obtained to this set of instances is close to 1230 s. The instance that takes longer to obtain the final solution is *mval9D* and the one that obtains faster the final solution is *mval2B*.

5.2 *lpr*-Instances

Some of the characteristics referred for the *mval* instances can be also applied to the *lpr* instances, like:

- All the vehicles begin and end their trips on a special node, called depot.

- When servicing an arc/edge it will be served just one time by that vehicle.
- The deadheading cost and the service cost of each arc/edge will be different from instance to instance

The main differences between these two types of instances lie on the characteristics listed beneath.

- The disposal cost, 300, is charged on this set of instances.
- The objective function of the *lpr* instances is given by the sum of the three different costs: deadheading cost, service cost and disposal cost.
- The instances *lpr-a* and *lpr-b* are composed for more required arcs than required edges while the instances *lpr-c* contains more required edges than required arcs.

These instances have a larger number of nodes and of required arcs/edges. This information can be observed in Table 5.5.

Table 5.5: Characteristics of the *lpr* instances

File	$\|N\|$	$\|A \cup E\|$	$\|A_r\|$	$\|E_r\|$
lpr-a-01	28	94	52	0
lpr-a-02	53	169	99	5
lpr-a-03	146	469	271	33
lpr-a-04	195	651	469	34
lpr-a-05	321	1056	748	58
lpr-b-01	28	63	45	5
lpr-b-02	53	117	92	9
lpr-b-03	163	361	279	26
lpr-b-04	248	582	493	8
lpr-b-05	401	876	764	37
lpr-c-01	28	52	11	39
lpr-c-02	53	101	23	77
lpr-c-03	163	316	61	241
lpr-c-04	277	604	142	362
lpr-c-05	369	841	416	387

Like the first Table presented in the section about the results obtained to the *mval* instances, this table contains a column to present the number of the nodes (column 2) of the respective instance, a column with information about the total required arcs plus required edges (column 3), and two separated columns comprising information about required arcs (column 4) and required edges (column 5).

The same approach used to solve the *mval* instances was followed to solve the *lpr* instances.

After the specification of the main principles, the results found to this set of instances are exposed in the next sections.

5.2.1 Phase1 - Mega-Route

The results obtained by this phase are shown in Table 5.6.

Table 5.6: Phase 1 - Results obtained to *lpr* Instances

Phase 1 - Mega-Route		
File	**Nodes**	**Objective Function**
lpr-a-01	28	13484
lpr-a-02	53	27666
lpr-a-03	146	74367
lpr-a-04	195	122732
lpr-a-05	321	<u>193915</u>
lpr-b-01	28	14474
lpr-b-02	53	28483
lpr-b-03	163	75792
lpr-b-04	248	<u>122624</u>
lpr-b-05	401	<u>200850</u>
lpr-c-01	28	18295
lpr-c-02	53	<u>35152</u>
lpr-c-03	163	<u>106822</u>
lpr-c-04	277	<u>164058</u>
lpr-c-05	369	<u>250365</u>

Column 2 shows the number of nodes of the respective instance and column 3 the value obtained for the objective function to the different tested instances.

It was specified a time limit to CPLEX software provide the final solution of this phase. If this maximum time is attained before the optimal solution is reached the methods used by this software should stop and return the solution found at that moment.

Observing Table 5.6 it's noticed that in some values obtained for some instances for the objective function are underlined. These values represent the instances where the maximum time (one hour) specified previously was reached.

5.2.2 Phase2 - Intermediate Solution

Table 5.7 contains the results obtained to the Phase 2 of the approach followed by this work.

Table 5.7: Phase 2 - Results obtained to *lpr* Instances

Phase 2 - Intermediate Solution			
File	**Objective Value**	**Total Demand**	**Number of Vehicles**
lpr-a-01	13591	11235	2
lpr-a-02	29457	23446	3
lpr-a-03	81430	64709	7
lpr-a-04	136283	108635	11
lpr-a-05	221676	170514	18
lpr-b-01	14950	12142	2
lpr-b-02	29575	23312	3
lpr-b-03	83000	63624	7
lpr-b-04	137664	103770	11
lpr-b-05	229634	171408	18
lpr-c-01	18927	16662	2
lpr-c-02	37581	31718	4
lpr-c-03	119294	97917	10
lpr-c-04	179548	149531	15
lpr-c-05	280251	227186	23

The aforementioned table is composed by four different columns. The first one represents the name of the respective instance, and the others contain information about the value obtained to the objective function (column 2), total demand that needs to be collected in the respective instance (column 3) and the number of available vehicles (column 4).

5.2.3 Phase3 - Final Solution

The final results obtained to this set of instances are presented in Table 5.8.

Table 5.8: Phase 3 - Results obtained to *lpr* Instances

Phase 3 - Final Solution				
File	**Best known value**	**Objective Value**	**Time(sec)**	**Gap(%)**
lpr-a-01	13484*	13554	312	0,52%
lpr-a-02	28052*	29300	1076	4,45%
lpr-a-03	76039	80665	5121	6,08%
lpr-a-04	126941	135280	8355	6,57%
lpr-a-05	202736	219715	18525	8,37%
lpr-b-01	14835*	14950	489	0,78%
lpr-b-02	28654*	29384	527	2,55%
lpr-b-03	77821	82693	5051	6,26%
lpr-b-04	126754	137277	11547	8,30%
lpr-b-05	209791	227983	53836	8,67%
lpr-c-01	18639*	18897	377	1,38%
lpr-c-02	36255*	37472	4330	3,36%
lpr-c-03	109980	118233	8196	7,50%
lpr-c-04	168441	178755	14591	6,12%
lpr-c-05	257890	278993	54467	8,18%
Average			12453,33	5,27%
Maximum			54467	8,67%
Minimum			312	0,52%

This table contains information about the best known value depicted on [18] (column 2), the value obtained by the application of the approach during the realization of this work (column 3), the CPU time in seconds (column 4) and finally the gap (column 5).

It's important to mention that if the best known value of an instance is the optimal solution it is marked with an (*).

Results Discussion

In spite of to some instances (*lpr-a-05, lpr-b-04, lpr-b-05, lpr-c-02, lpr-c-03, lpr-c-04, lpr-c-05*) the values obtained may or may not be the optimal solution to the Mega-Route of the respective instance the different values obtained by this phase of the methodology are lower than the best known results attained by [18].

In what concerns this set of instances the values obtained to the Intermediate Solution (Phase 2 of the approach developed during this work) are much higher than the best known result of the respective instance. The Metaheuristic (Phase 3) used by next phase will guarantee that the final solutions obtained are better than the solution given by the previous phase. If the value obtained to the Intermediate solution is far from the optimal solution, the Tabu Search may not find a value near to the optimal value of the respective instance.

The values obtained to the final solution of this set of instances are close to the best known value obtained by other authors. These values are between 0.52% and 8.67%.

The instance where this approach was more effective is the instance called *lpr-a-01* with a value registered to the gap close to 0.5%. The worst scenario verified was acquired for the instance *lpr-b-05* with a gap value near to 9%.

The average of the values obtained to all the instances of this set is close to 5%. This means that the solutions obtained are near to the best known results to the same instances studied by other authors, guaranteeing that this approach when applied to a set of complex instances, will be able providing good final results.

The main disadvantage verified to the more complex instances lies on the time demanded to find the final solution. To these instances the average time calculated is close to 9500 seconds. This time will increase with the complexity of the problem under study and that's the reason why in the *lpr* instances the time needed to obtain the final solutions is much higher than the one obtained to the *mval* instances.

5.3 Summary

This chapter intends to show the results obtained for two different sets of instances. The first set, called *mval* instances, comprises the simplest instances that contain (24-50) nodes and (43-138) links and all of them are required. The second set, designated *lpr* instances, is composed by complex instances with (28-401) nodes and (40-1056) links where just a subset of these are required.

To evaluate the quality of the approach followed in this work should be observed two different parameters:

1. The difference between the value obtained to the objective function by this approach and the value obtained by other authors

2. The time needed to find the final solution by the approach.

To the *mval* instances the average value obtained to the gap was near to 15% while to the *lpr* instances this value is near to 5%. Concerning the average obtained to the CPU time the *lpr* instances take much longer (12453 seconds) to obtain the final solution than the *mval* instances (1230 seconds). This happens due to the complexity of the instances that belongs to each set.

In general, the approach used in this work obtains good results to the instances under test but when it is applied to complex problems this will need much time to obtain the final solution.

Chapter 6

Conclusions and Future Work

This chapter intends to expose the main conclusions that can be taken from this Master Thesis and the work that can be realized in the future.

6.1 Conclusions

The problem approached during this work was a MCARP (Mixed Capacitated Arc Routing Problem). This kind of problems is described by a mixed graph where some of its arcs have a demand associated to it (required arcs). This type of problems aims the obtention of a solution minimizing its cost and containing all the required arcs.

The biggest obstacle verified during this project was the approaching of the problem as a MCARP, since in the literature most of the works are solved as a NRP (Node Routing Problem). One approach used by a large number of authors is the transformation of an Arc Routing Problem into a Node Routing Problem.

To solve the problem proposed in this work two different Optimization Models were studied. These models were focus in the work developed by [18]. The model used in this work was built based on these two different models. Through the study of these models were obtained a better knowledge and sensitivity about the restrictions associated with this kind of problems. This knowledge allowed to the author a deep comprehension about the difficulty of solving this type of problems in real life situations due to its complexity and its possible different characteristics.

The methodology used was developed in three different stages where each one has a different aim.

The approach developed is composed by different methods used in optimization problems. The first phase uses CPLEX to apply Exact Methods providing an infeasible optimal solution that is used as an input in the next phase. The second stage uses Heuristic Methods in order to give other different solution, a feasible one, with different characteristics and a higher value of the objective function. In the last phase a Metaheuristic is applied that intends to increase the quality of the solution obtained by the previous stage.

After the development of the aforementioned approach, it was applied to two different set of instances.

To the first set, the *mval* instances, the results obtained were not much near to the best known values but the time needed to obtain that solution were not very high.

Concerning the second set, *lpr* instances, this approach will take much more time than to the *mval* instances. That happens due to the complexity of this group of problems. The time needed to obtain a final solution will increase with the complexity of the problem under study. Despite this inconvenient the final results obtained to this second set of instances were better than the ones attained to the previous set.

Despite the time demanded to obtain a solution is an important parameter to evaluate this quality index of a routing problem solver, is not absolutely required that this time should be low, but since the real life routing problem does not need to be changed in a regular way, the weight of this parameter can be minimized.

The greatest contribution of the work developed during this Master Thesis lies on the development of a new approach to solve similar problems to the one solved in this work since the adopted methodology was capable of providing satisfactory results to a large number of tested instances in an acceptable computational time.

6.2 Future Work

To continue this work will be interesting the application of two different procedures. Next will be exposed this two possible modifications to the applied approach.

To obtain better results to the tested instances would be interesting to implement a Hybrid Method in the last stage of the methodology and compare the results attained by the one implemented in this Master Thesis.

A Hybrid Method is a combination of Exact Methods and Aproximative Methods. This combination intends to gain with the advantages of these two different methods.

Concerning the work developed by the author, a hybrid method should be applied during the Metaheuristic phase. The exact methods are used by CPLEX when the reconstruction of a route is performed. This procedure will be performed in each iteration of the method since the improvement heuristic, heuristic 2-opt, is used to change the disposition of the required arcs that composes the different routes of a solution. Applying it the solution obtained to the changed route could be better than the one that is actually obtained since this last one may or may not be the optimal one. If this was the procedure followed in this work would be interesting verify the following actually happens:

- Results would be obtained faster since the algorithm already developed takes too long to reconstruct the changed routes;

- In many of the realized iterations, the solutions obtained would be better than the ones now attained since to all of them the routes that compose them were constructed by exact methods that provide the optimal solution.

Other modification that may be performed in order to verify the results that could be obtained is the using of multi-tasks in the algorithm developed. This modification could be applied in the way that the routes were constructed, since the algorithm is applied to each route at a time. This would enable the construction of all the routes that forms the solution at the same time. If this was implemented could be done more tests in the instances with a large number of iterations without increase significantly the computational time needed to obtain the final solution.

References

[1] Tolga Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, June 2006.

[2] José-Manuel Belenguer, Enrique Benavent, Philippe Lacomme, and Christian Prins. Lower and upper bounds for the mixed capacitated arc routing problem. *Computers & Operations Research*, 33(12):3363–3383, December 2006.

[3] Dimitris Bertsimas and David Simchi-levi. A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. *Operations Research*, 44:286–304, 1993.

[4] G. Clarke and J. W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581, 1964.

[5] William Cook. The traveling salesman problem, acessed in january 2, 2012, http://www.tsp.gatech.edu/index.html.

[6] Angel Corbera, Rafael Martmh, Eulalia Martmh, and David Soler. The Rural Postman Problem on mixed graphs with turn penalties. *Computers & Operations Research*, 29, 2002.

[7] Jean-François Côté and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research*, 198(2):464–469, October 2009.

[8] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[9] Agência Portuguesa do Ambiente. Plano Estratégico para os Resíduos Sólidos Urbanos 2007-2016 (PERSU II), 2007.

[10] Agência Portuguesa do Ambiente. Caracterização da situação actual – 2010/2011. 2011.

[11] Dreo, A. Pétrowski, P. Siarry, and E. Taillard. *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer, 1 edition, December 2005.

[12] M. Dror. *Arc routing: theory, solutions, and applications*. Kluwer Academic, 2000.

[13] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. Arc Routing Problems, Part I: The Chinese Postman Problem. *Operations Research*, 43(2):231–242, 1995.

[14] H. A. Eiselt, Michel Gendreau, and Gilbert Laporte. Arc Routing Problems, Part II: The Rural Postman Problem. *Operations Research*, 43(3):399–414, 1995.

[15] José Soeiro Ferreira. Grafos e Redes, 2008.

[16] Richard Y K Fung, Jiafu Tang, and Jun Zhang. A Multi-Depot Vehicle Routing Problem with Weight-Related Costs. *Manufacturing Engineering*, pages 1028–1033, 2009.

[17] Edward A. Golden, Bruce L.; Raghavan, S.; Wasil. *THE VEHICLE ROUTING PROBLEM : LATEST ADVANCES AND NEW CHALLENGES*.

[18] Luís Gouveia, Maria Cândida Mourão, and Leonor Santiago Pinto. Lower bounds for the mixed capacitated arc routing problem. *Computers & Operations Research*, 37(4):692–699, April 2010.

[19] F.S. Hillier and G.J. Lieberman. *Introduction to operations research*. McGraw-Hill International Editions. McGraw-Hill, 2001.

[20] R. Horst, P.M. Pardalos, and N.V. Thoai. *Introduction to global optimization*. Nonconvex optimization and its applications. Kluwer Academic Publishers, 2000.

[21] Hu Huang, Hong-Bo Xie, Jing-Yi Guo, and Hui-Juan Chen. Ant colony optimization-based feature selection method for surface electromyography signals classification. *Computers in biology and medicine*, 42(1):30–8, January 2012.

[22] L. Jourdan, M. Basseur, and E.G. Talbi. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, 199(3):620–629, December 2009.

[23] Ismail Karaoglan, Fulya Altiparmak, Imdat Kara, and Berna Dengiz. A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery. *European Journal of Operational Research*, 211(2):318–332, June 2011.

[24] K.H. Rosen, J.G. Michaels, J.L. Gross, J.W. Grossman, and D.R. Shier. *Handbook of Discrete and Combinatorial Mathematics*. CRC Press LLC, USA, 1 edition, 2000.

[25] GLPK (GNU Linear Programming Kit). `http://www.gnu.org/software/glpk/`. acessed in January 24 , 2012.

[26] Alexandre Magrinho, Filipe Didelet, and Viriato Semiao. Municipal solid waste disposal in portugal. *Waste Management*, 26(12):1477 – 1489, 2006.

[27] W. Michiels, E.H.L. Aarts, and J. Korst. *Theoretical aspects of local search*. Monographs in theoretical computer science. Springer, 2007.

[28] T Nuortio, J Kytojoki, H Niska, and O Braysy. Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications*, 30(2):223–232, February 2006.

[29] Jacques Renaud, Fayez F Boctor, and Gilbert Laporte. Perturbation heuristics for the pickup and delivery traveling salesman problem. *Computers & Operations Research*, 29, 2002.

[30] Rodrigues, Ana Maria; Ferreira, José Soeiro. Rural Postman and related Arc Routing Problems. 2011.

[31] IBM ILOG CPLEX Optimization Studio. `http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/`. acessed in January 26 , 2012.

[32] G.D. Taylor. *Logistics engineering handbook*. CRC Press, 2008.

[33] Shigeru Tsubakitani and James R. Evans. Optimizing tabu list size for the traveling salesman problem. *Comput. Oper. Res.*, 25:91–97, February 1998.