

**MESTRADO EM**  
MÉTODOS QUANTITATIVOS PARA A DECISÃO  
ECONÓMICA E EMPRESARIAL

**TRABALHO FINAL DE MESTRADO**  
TRABALHO DE PROJETO

OTIMIZAÇÃO DE ITINERÁRIOS PARA A FISCALIZAÇÃO DE  
ESTACIONAMENTO

RICARDO SALVADOR GOMES MONTEIRO PEREIRA

NOVEMBRO-2020

**MESTRADO EM**  
MÉTODOS QUANTITATIVOS PARA A DECISÃO  
ECONÓMICA E EMPRESARIAL

**TRABALHO FINAL DE MESTRADO**

TRABALHO DE PROJETO

OTIMIZAÇÃO DE ITINERÁRIOS PARA A FISCALIZAÇÃO DE  
ESTACIONAMENTO

RICARDO SALVADOR GOMES MONTEIRO PEREIRA

**ORIENTAÇÃO:**

PROF.<sup>a</sup> DOUTORA MARIA CÂNDIDA VERGUEIRO MONTEIRO CIDADE MOURÃO  
PROF.<sup>a</sup> DOUTORA MARIA MARGARIDA DE OLIVEIRA MOZ CARRAPA

NOVEMBRO-2020

# DEDICATÓRIA

*À minha mãe*

## AGRADECIMENTOS

---

Antes de mais, gostaria de expressar a minha sincera gratidão ao projeto de bolsas de investigação para licenciados do Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, CMAFcIO.

Pela orientação, o apoio e o incentivo demonstrado ao longo deste projeto gostaria também de expressar a minha gratidão e apreço à *Prof.<sup>a</sup> Doutora Maria Cândida Vergueiro Monteiro Cidade Mourão* e à *Prof.<sup>a</sup> Doutora Maria Margarida de Oliveira Moz Carrapa*.

Em conclusão, agradeço também a minha família e aos meus amigos por todo o apoio incondicional, pela paciência e pela compreensão nesta etapa decisiva do meu percurso académico.

## RESUMO

---

Ao longo dos últimos anos os problemas de roteamento nos arcos têm vindo a ser estudados com grande intensidade. Neste tipo de problemas, o objetivo é atravessar determinadas ligações, habitualmente relacionadas com as ruas ou vias, representadas num grafo que, no presente trabalho, se integram em zonas de estacionamento na cidade de Lisboa, geridas pela EMEL (Empresa Municipal de Mobilidade e Estacionamento de Lisboa). A ideia central consiste em construir percursos de trabalho para cada fiscal de estacionamento que permitam atender todas as necessidades de fiscalização de estacionamento da melhor maneira possível. Este projeto surge na sequência de um estudo realizado por Ferreira (2016) e visa dar seguimento a esta investigação introduzindo uma metaheurística com resultados promissores.

Os veículos (fiscais) encontram-se inicialmente no depósito. Eles iniciam os seus percursos de modo a fiscalizar as ligações especificadas e, de seguida, retornam ao depósito, respeitando a capacidade. Neste caso, a restrição de capacidade de cada veículo corresponde à duração do turno de cada fiscal. Para avaliar a necessidade de fiscalização de cada rua foi introduzido um parâmetro, denominado por criticidade, que varia de acordo com a hora do dia. Assim, o objetivo do problema assenta na maximização da criticidade total associada a todos os percursos.

Este estudo pretende apresentar uma estratégia de solução aproximada para o Problema de Otimização de Itinerários para a Fiscalização de Estacionamento (POIFE) e realizar uma análise dos resultados computacionais da metodologia desenvolvida. São propostas uma heurística construtiva para obtenção de soluções admissíveis iniciais e uma abordagem metaheurística, baseada em *Tabu Search* (TS), para resolver instâncias de grande dimensão. Esta, por sua vez, inclui uma heurística melhorativa de pesquisa local, *2-opt*. Os algoritmos propostos foram implementados com recurso ao *Microsoft Excel Visual Basic for Applications* e os testes relativamente ao seu desempenho foram realizados em pequenos exemplos gerados aleatoriamente e também em instâncias da vida real baseadas em dados de ruas de Lisboa.

**Palavras-Chave:** Fiscalização de Estacionamento; Problemas de Roteamento nos Arcos; Heurísticas; Metaheurística; Tabu Search

## ABSTRACT

---

Over the past years, arc routing problems have been studied intensively. In this type of problems, the main purpose is to cross certain connections that are related to streets or roads of a graph, which, in this case, represent the parking lot areas in the city of Lisbon, managed by EMEL (Empresa Municipal de Mobilidade e Estacionamento de Lisboa). The main idea is to build paths for each parking enforcement officer that may attend all the inspection needs in the best way possible. This project follows a study by Ferreira (2016) and aims to continue the investigation introducing a metaheuristic with promising results.

Vehicles (officers) are initially in the depot. They start their routes in order to inspect the requested links and then return to the depot taking into account the capacity. The capacity constraint is related to the duration of the officers' shifts. In order to assess the inspection needs of each area, a parameter was assigned, called criticality, which changes throughout the day. Therefore, the objective of the problem is to maximize the total criticality of all routes.

This study intends to present an approximate solution strategy for the route optimization problem for parking enforcement (ROPPE) and to perform an analysis of the computational results of the developed methodology. A constructive heuristic to obtain initial feasible solutions, and a metaheuristic approach based on a Tabu Search (TS) for solving large instances are proposed. TS includes an improving local search heuristic, 2-opt. The proposed algorithms were implemented using Microsoft Excel Visual Basic for Applications and their performance were tested through randomly generated examples and also real life instances based data from streets of Lisbon.

**Keywords:** Parking Enforcement; Arc Routing Problems; Heuristics; Metaheuristic; Tabu Search

# ÍNDICE GERAL

<b>DEDICATÓRIA .....</b>	<b>I</b>
<b>AGRADECIMENTOS .....</b>	<b>II</b>
<b>RESUMO.....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>IV</b>
<b>ÍNDICE GERAL.....</b>	<b>V</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>VII</b>
<b>ÍNDICE DE TABELAS.....</b>	<b>VIII</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>IX</b>
<b>CAPÍTULO I: INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1 Introdução .....</b>	<b>1</b>
<b>1.2 EMEL .....</b>	<b>2</b>
<b>1.3 Estrutura do Relatório .....</b>	<b>3</b>
<b>CAPÍTULO II: ENQUADRAMENTO DO PROBLEMA .....</b>	<b>4</b>
<b>2.1 Definição do Problema .....</b>	<b>4</b>
<b>2.2 Criticidade .....</b>	<b>5</b>
<b>2.3 Fiscalização.....</b>	<b>6</b>
<b>CAPÍTULO III: REVISÃO DE LITERATURA.....</b>	<b>9</b>
<b>CAPÍTULO IV: METODOLOGIA.....</b>	<b>12</b>
<b>4.1 Heurística Construtiva Aleatória (HCA) .....</b>	<b>12</b>
4.1.1 Introdução .....	12
4.1.2 Pseudocódigo HCA .....	13
4.1.3 Exemplo de Aplicação da HCA.....	15
<b>4.2 Tabu Search (TS).....</b>	<b>19</b>
4.2.1 Apresentação da Metaheurística .....	19
4.2.2 Descrição Geral do Pseudocódigo TS .....	20

<b>4.3 Heurística 1-opt .....</b>	<b>23</b>
4.3.1 Introdução .....	23
4.3.2 Descrição do Pseudocódigo Heurística 2-opt.....	24
<b>CAPÍTULO V: ANÁLISE COMPARATIVA DE RESULTADOS.....</b>	<b>26</b>
<b>5.1 Teste Computacional 1.....</b>	<b>26</b>
<b>5.2 Teste Computacional 2.....</b>	<b>27</b>
<b>CAPÍTULO VI: CONCLUSÕES.....</b>	<b>29</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>30</b>
<b>ANEXOS.....</b>	<b>32</b>
<b>Anexo 1 – Rede.....</b>	<b>32</b>
<b>Anexo 2 – Soluções Admissíveis (Teste 1).....</b>	<b>33</b>
<b>Anexo 3 – Soluções Admissíveis (Teste 2).....</b>	<b>34</b>



## ÍNDICE DE FIGURAS

---

Figura 1 – Ruas com estacionamento em um único lado .....	7
Figura 2 – Ruas com dois sentidos e estacionamento em ambos lados.....	7
Figura 3 – Ruas com estacionamento nos dois lados e na placa central .....	8
Figura 4 – Ruas com um sentido e estacionamento em ambos os lados .....	8
Figura 5 – Pseudocódigo da Heurística Construtiva Aleatória.....	14
Figura 6 – Legenda para os exemplos .....	15
Figura 7 – Rede do exemplo .....	16
Figura 8 – Primeiro percurso do AFE 1 .....	18
Figura 9 – Segundo percurso do AFE 1 .....	18
Figura 10 - Primeiro percurso do AFE 2 .....	18
Figura 11 - Segundo percurso do AFE 2 .....	18
Figura 12 – Pseudocódigo da metaheurística Tabu Search .....	22
Figura 13 - Percurso $P_0$ na solução corrente.....	24
Figura 14 - Percurso $P'$ obtido pela heurística 2-opt .....	24
Figura 15 – Pseudocódigo da Heurística 2-opt.....	25
Figura 16 - Criticidade média teste 1.....	28
Figura 17 – Criticidade média teste 2.....	28
Figura 18 – Rede com quatro zonas de fiscalização (Fonte: Ferreira, 2016) .....	32

## ÍNDICE DE TABELAS

---

Tabela 1 – Terminologia utilizada na HCA.....	13
Tabela 2 – Exemplo HCA .....	17
Tabela 3 – Terminologia utilizada no TS .....	20
Tabela 4 – Valores para os parâmetros.....	26
Tabela 5 - Resultados médios para o teste 1 .....	27
Tabela 6 - Resultados médios para o teste 2.....	28
Tabela 7 – Resultados solução admissível 1 (T1) .....	33
Tabela 8 - Resultados da solução admissível 2 (T1) .....	33
Tabela 9 - Resultados da solução admissível 3 (T1) .....	33
Tabela 10 - Resultados da solução admissível 4 (T1) .....	34
Tabela 11 - Resultados para a solução admissível 5 (T1) .....	34
Tabela 12 - Resultados para a solução admissível 1 (T2) .....	34
Tabela 13 - Resultados para a solução admissível 2 (T2) .....	35
Tabela 14 - Resultados para a solução admissível 3 (T2) .....	35
Tabela 15 - Resultados para a solução admissível 4 (T2) .....	35
Tabela 16 - Resultados para a solução admissível 5 (T2) .....	36

## LISTA DE ABREVIATURAS

---

AFE – Agente de Fiscalização de Estacionamento

ARP – *Arc Routing Problem*

ARPP – *Arc Routing Problems with Profits*

CARP – *Capacitated Arc Routing Problem*

CMC – Caminho Mais Curto

CML – Câmara Municipal de Lisboa

CPP – *Chinese Postman Problem*

EMEL – Empresa Municipal de Mobilidade e Estacionamento de Lisboa

GRASP – *Greedy Randomized Adaptive Search Procedure*

ISEG – Instituto Superior de Economia e Gestão

MBCPP – *Maximum Benefit Chinese Postman Problem*

MQDEE – Métodos Quantitativos para a Decisão Económica e Empresarial

MRPP – *Mixed Rural Postman Problem*

PCRPP – *Prize-Collecting Rural Postman Problem*

POIFE – Problema de Otimização de Itinerários para a Fiscalização de Estacionamento

PTP – *Profitable Tour Problem*

ROPPE – *Route Optimization Problem for Parking Enforcement*

RPP – *Rural Postman Problem*

SA – Solução Admissível

TOARP – *Team Orienteering Arc Routing Problem*

VBA – *Visual Basic for Applications*

## CAPÍTULO I: INTRODUÇÃO

---

### 1.1 Introdução

O presente trabalho foi iniciado no âmbito de uma Bolsa de Investigação para Licenciados decorrente do projeto de I&D do Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, CMAFcIO, Universidade de Lisboa. A bolsa, durante o período decorrido entre 2 de setembro e 31 de dezembro de 2019, permitiu aplicar os conhecimentos adquiridos ao longo do Mestrado em Métodos Quantitativos para a Decisão Económica e Empresarial (MQDEE) e dar início ao projeto de Trabalho Final de Mestrado.

Com o decorrer dos anos foram realizados vários estudos de modo a identificar soluções que melhorassem a mobilidade e o estacionamento na cidade de Lisboa. O grande impacto deste na qualidade de vida na cidade exige uma grande atenção aos fatores que contribuem para a sua melhoria. A atividade desenvolvida pela EMEL tem um papel importante no estacionamento, na medida em que é a entidade que efetua a fiscalização dos diferentes tipos de infração que ocorrem e que perturbam a mobilidade na cidade.

Este projeto visa contribuir para melhorar as condições de mobilidade urbana e segurança rodoviária na cidade de Lisboa. Pretende-se otimizar os percursos realizados pelos Agentes de Fiscalização de Estacionamento (AFE) da EMEL, que se deslocam a pé, tendo em conta os respetivos horários de trabalho e a necessidade de fiscalização das diferentes zonas de estacionamento. Adicionalmente, considera-se que o Problema da Otimização de Itinerários para a Fiscalização de Estacionamento (POIFE) pode ser encarado como um *Arc Routing Problem*.

Nas últimas décadas, houve uma grande evolução no desenvolvimento de metodologias, tanto exatas como aproximadas, para problemas de roteamento. No entanto, algoritmos exatos não são capazes de resolver, de um modo consistente, problemas de grandes dimensões, e, por isso, são estudadas outras abordagens, como heurísticas e metaheurísticas. Heurísticas são algoritmos que permitem identificar soluções para problemas específicos, em tempo considerado adequado, mas não garantindo a otimalidade das soluções. Metaheurísticas representam abordagens baseadas em heurísticas que iterativamente tentam chegar a soluções de melhor qualidade em

problemas difíceis de otimização. São metodologias gerais que têm como objetivo ultrapassar os obstáculos enfrentados por certas heurísticas, nomeadamente, de pesquisa local que, muitas vezes, tendem a terminar em ótimos locais de fraca qualidade. Ambos os métodos podem ser utilizados para determinar uma solução aproximada para um problema de otimização. A metaheurística desenvolvida neste projeto permite, com pouca exigência computacional, obter rotas que percorrem as ruas da cidade onde é mais suscetível que ocorram incumprimentos no estacionamento, facilitando o trabalho dos fiscais da EMEL.

Durante este projeto foi necessário primeiramente identificar e descrever o problema a tratar. De seguida, deu-se início a uma pesquisa bibliográfica que consistiu essencialmente na leitura, seleção e arquivamento de vários artigos científicos considerados relevantes e que permitiu, por sua vez, escolher uma metodologia de investigação baseada em métodos heurísticos. Escolhida a metodologia a utilizar procedeu-se ao seu desenvolvimento, isto é, à sua escrita em pseudocódigo e posteriormente à implementação com recurso ao *Microsoft Excel Visual Basic for Applications* e à realização de testes computacionais.

## **1.2 EMEL**

A gestão do estacionamento em Lisboa é da competência da EMEL (EMEL, 2017), Empresa Municipal de Mobilidade e Estacionamento de Lisboa, autoridade que lhe é atribuída pelo art. 5º, do DL nº44/2005, de 23 de Fevereiro. A empresa leva a cabo a prestação de serviços de interesse geral no âmbito da gestão, da mobilidade urbana e do desenvolvimento de novas estratégias que permitam uma melhor fiscalização do estacionamento em Lisboa, tendo em vista a contribuição para uma melhor mobilidade e segurança rodoviária.

A EMEL procura gerir e ordenar o estacionamento no município de Lisboa, de modo a facilitar a mobilidade e a proporcionar melhores condições de bem-estar para quem vive e circula na cidade. A empresa desempenha um papel ativo no ordenamento do espaço público, na medida em que visa adequar a oferta de estacionamento às necessidades da procura. Por outro lado, a EMEL procura, ao longo do exercício da sua atividade, consolidar a sua posição como uma empresa de referência, tanto a nível nacional como internacional, nas áreas de estacionamento e mobilidade urbana.

A fiscalização do estacionamento urbano público em Lisboa é exercida pelos Agentes de Fiscalização de Estacionamento (AFE) da EMEL. Estes, deslocando-se a pé, verificam o cumprimento das normas de estacionamento e também o estado de conservação e funcionamento dos equipamentos e da sinalização existentes nas vias sob jurisdição municipal.

### **1.3 Estrutura do Relatório**

A estrutura deste documento espelha as diferentes fases do projeto. Após uma breve introdução no capítulo 1, sobre a motivação que deu origem à elaboração deste projeto, os seus principais objetivos e sobre a empresa em causa, este relatório estende-se por mais quatro capítulos distintos, aos quais se acrescentam, no final, as referências e os anexos. No capítulo 2, dá-se início ao enquadramento teórico do problema, de modo a contextualizar a metodologia de investigação utilizada, baseada em heurísticas. No capítulo 3, realiza-se uma revisão bibliográfica relativamente à temática em estudo. De seguida é apresentada, no capítulo 4, a metodologia desenvolvida para fazer face ao problema descrito nos capítulos anteriores. Neste capítulo são descritos os procedimentos implementados em *VBA*, incluindo o desenvolvimento de uma heurística construtiva, uma heurística melhorativa e uma metaheurística *Tabu Search*. Posteriormente, procedeu-se a testes computacionais que permitiram comparar os resultados das heurísticas desenvolvidas, evidenciando-se as limitações e, por fim, as considerações finais do trabalho nos capítulos 5 e 6, respetivamente.

## **CAPÍTULO II: ENQUADRAMENTO DO PROBLEMA**

---

### **2.1 Definição do Problema**

O município de Lisboa encontra-se organizado em várias zonas e as suas ruas são classificadas em segmentos e em conetores. Os segmentos são troços de ruas que os fiscais da EMEL devem percorrer sem interrupção para verificar a ocorrência de irregularidades no estacionamento e os conetores servem para garantir a ligação entre os segmentos. Por vezes, pode ser necessário atravessar certos segmentos sem efetuar fiscalização, ou seja como conetores.

Tal como em Ferreira (2016), para auxiliar os processos de fiscalização foi utilizado um parâmetro, a criticidade, que permite avaliar quais os segmentos e zonas com maior necessidade de fiscalização. Em cada hora de exploração, é associado a cada segmento um dado valor de criticidade influenciado pelo historial de ocupação do segmento nessa hora, pelo tempo decorrido desde a última fiscalização e por um histórico relativamente ao número e à gravidade das denúncias emitidas no segmento. É de salientar ainda que um segmento pode ser fiscalizado apenas uma única vez em cada período de duas horas. Após esse período, o segmento pode ser novamente fiscalizado, eventualmente com um novo valor de criticidade.

O objetivo deste trabalho advém da necessidade de otimizar os percursos a serem realizados pelos AFE, que se deslocam a pé, respeitando o número limitado de agentes e também um conjunto de restrições adicionais como, por exemplo, as janelas de tempo relativas aos horários de trabalho dos agentes.

Nas zonas de estacionamento de duração limitada, o estacionamento fica sujeito ao pagamento de uma tarifa de utilização de segunda a sexta, das 9 às 19 horas, e aos sábados das 9 às 13. No entanto, cada AFE trabalha em média 7 horas por dia, em dois turnos, a que correspondem dois percursos distintos, um a ser realizado no turno da manhã e o outro durante a tarde, após a paragem para almoço. Para poder cobrir o tempo diário de exploração das 9 às 19 horas, nos dias úteis, foram criados dois horários, um que começa às 9 (das 9 às 13 e das 14 às 17) e outro às 11 (das 11 às 14 e das 15 às 19).

No início de cada turno, cada AFE parte da central da EMEL, o depósito, e de seguida segue para a zona onde irá iniciar a fiscalização. A fiscalização numa zona tem início e fim em locais denominados por pontos base que correspondem a locais com

paragens de transportes públicos em ligação com a central da EMEL. Um percurso atribuído a um AFE tem assim início no depósito, com a ligação a um ponto base, inclui segmentos e conetores e termina com outra ligação entre um ponto base e o depósito, não sendo necessariamente o inicial. Para o cálculo da duração de um percurso deve-se ter em consideração o tempo que um AFE necessita para percorrer os segmentos e os conetores (incluindo as ligações entre o depósito e os pontos base), bem como o tempo para fiscalizar cada segmento.

Em resumo, pretende-se determinar dois percursos diários de fiscalização para cada AFE, maximizando uma função que reflete a necessidade de fiscalização dos segmentos e que respeite as seguintes restrições:

- O número limitado de AFE disponíveis;
- Cada AFE inicia o respetivo turno partindo do depósito e regressando ao mesmo no final;
- Cada percurso tem início e fim em pontos base com ligação ao depósito;
- Um segmento só pode ser fiscalizado uma única vez a cada duas horas sendo definidos, para este efeito, períodos de uma hora (60 minutos);
- Um segmento não pode ser simultaneamente fiscalizado por mais do que um AFE;
- A fiscalização de cada segmento é realizada na totalidade, isto é, não se podem fiscalizar partes de segmentos;
- O horário de trabalho dos AFE deve ser respeitado.

Uma solução admissível para o problema de otimização em causa corresponde a dois percursos diários, por AFE, que verifiquem todas as restrições anteriormente referidas.

O POIFE é de difícil resolução, portanto, para o resolver, tal como em Ferreira (2016), optou-se por considerar uma abordagem heurística. São então propostas uma heurística construtiva para a obtenção de soluções admissíveis iniciais, uma metaheurística *Tabu Search* e uma heurística melhorativa de pesquisa local, *2-opt*.

## 2.2 Criticidade

Ao longo do dia verificam-se diferentes oscilações relativamente à procura de estacionamento. Portanto, a cada segmento pode ser associado um valor de criticidade



diferente em cada hora de exploração. Este valor é calculado no final de um dia para vigorar no dia seguinte. Tal como foi abordado em Ferreira (2016), para determinar a criticidade de um segmento foi necessário reunir um conjunto de fatores essenciais:

- **Ocupação de um segmento numa determinada hora** – retrata um número médio de veículos que estacionam num segmento numa determinada hora. Segmentos numa zona onde normalmente muitos veículos estacionam numa determinada hora devem ter maior criticidade associada, uma vez que se espera que a probabilidade de incumprimento nessa área seja grande.

- **Número de denúncias emitidas** – Se numa determinada hora do dia, há um elevado número de denúncias emitidas num segmento, a necessidade de fiscalizar esse segmento aumenta.

- **Tempo decorrido desde a última fiscalização** – não deve ser permitido que um segmento fique demasiado tempo sem fiscalização, pois isso pode aumentar a probabilidade de surgirem incumprimentos no estacionamento. Este fator permite, portanto, manter o sistema atualizado relativamente aos segmentos que são fiscalizados com menor frequência.

- **Gravidade das denúncias** – este fator é essencial para diferenciar a gravidade das infrações cometidas em cada segmento, dando maior prioridade aos segmentos onde se registam infrações graves.

## 2.3 Fiscalização

A fiscalização das diferentes áreas de estacionamento implica a definição de quatro tipos de ligações distintas na rede do problema (Ferreira, 2016):

- Tipo 1: As ligações do tipo 1 representam ruas com estacionamento em apenas um lado que tenham quer tráfego num único sentido (Figura 1a)), quer nos dois sentidos (Figura 1b)). Em ambos os casos a fiscalização deve ser feita no sentido do trânsito, enquanto o sentido oposto pode ser utilizado como conector. A Figura 1c) denota os arcos correspondentes na rede.

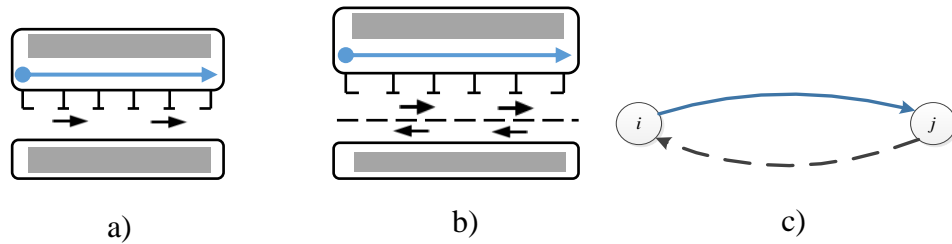


Figura 1 – Ruas com estacionamento em um único lado

- Tipo 2: As ligações do tipo 2 representam ruas com dois sentidos e estacionamento em ambos os lados, sendo a fiscalização realizada no sentido do trânsito. De salientar que a fiscalização de um lado não implica necessariamente a fiscalização do outro. O esboço da zona de estacionamento e os arcos correspondentes na rede encontram-se representados nas Figuras 2a) e 2b), respetivamente.

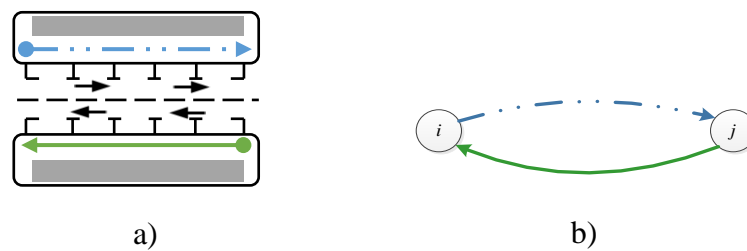


Figura 2 – Ruas com dois sentidos e estacionamento em ambos lados

- Tipo 3: Este tipo de ligação representa as ruas com estacionamento em ambos os lados e também na placa central que corresponde ao esboço da Figura 3a). A aresta na rede corresponde à Figura 3b). Esta aresta é substituída por dois arcos com direções opostas (Figura 3c)), sendo que a fiscalização em uma direção impede a fiscalização no sentido oposto. Considera-se que o arco  $(i, j)$  representa o caminho  $(i, j, i, j)$ , enquanto  $(j, i)$  representa o caminho  $(j, i, j, i)$  sendo, por isso, necessário a inspeção em apenas um dos sentidos.

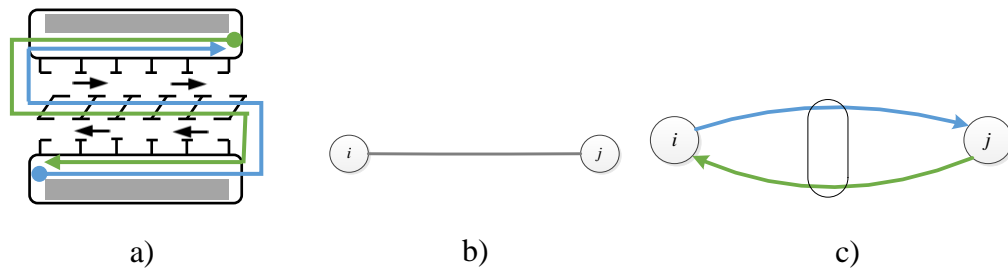


Figura 3 – Ruas com estacionamento nos dois lados e na placa central

- Tipo 4: Este tipo de ligação representa ruas com um único sentido e estacionamento nos dois lados. A Figura 4a) ilustra duas possibilidades de fiscalização. O AFE ou começa e acaba do lado esquerdo, ou começa e acaba do lado direito. A fiscalização nestas ruas deve ser feita no mesmo período e em ambos os lados, portanto, a fiscalização de um sentido implica também a fiscalização do sentido oposto. A Figura 4b) representa os dois arcos opostos que devem ser fiscalizados no mesmo período.

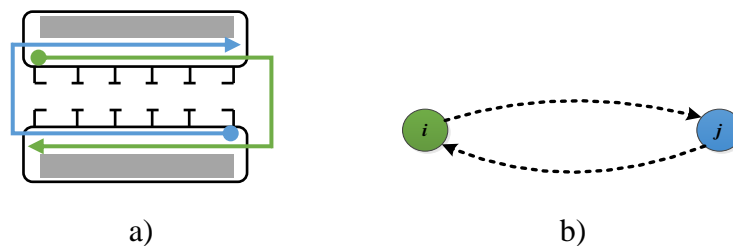


Figura 4 – Ruas com um sentido e estacionamento em ambos os lados

De modo a responder melhor à questão de investigação, o capítulo seguinte ilustra a literatura que permite evidenciar como a fiscalização do estacionamento pode ser modelada como um problema de otimização em redes, em particular, um problema de roteamento capacitado nos arcos.

## CAPÍTULO III: REVISÃO DE LITERATURA

---

A otimização de itinerários para a fiscalização de estacionamento pode ser modelada como um *Arc Routing Problem* (ARP). São exemplos de problemas de ARP, o *Chinese Postman Problem* (CPP), o *Rural Postman Problem* (RPP) e o *Capacitated Arc Routing Problem* (CARP).

O CPP foi pela primeira vez investigado pelo matemático chinês *Mei-ko Kwan* (1962). O conceito assenta na história de um carteiro chinês que pretendia entregar correspondência em todas as ruas de uma cidade percorrendo a menor distância total possível, ou seja, encontrar o percurso mais curto num grafo em que cada aresta fosse percorrida pelo menos uma vez. O CPP procura uma solução que permita cobrir todas as ruas, iniciando e regressando aos correios, com a menor distância total possível.

O problema do carteiro rural é uma generalização prática do problema do carteiro chinês e, tal como o nome indica, foi inicialmente associado a um problema de entrega de correio em áreas rurais. Corberán et al. (2000) descrevem uma abordagem para resolver o problema do carteiro rural em um grafo misto, *Mixed Rural Postman Problem* (MRPP), implementando um algoritmo *Tabu Search*. O objetivo consiste em determinar percursos de custo mínimo que atravessam, pelo menos uma vez, as ligações de um dado subconjunto de arcos e arestas. O RPP, contrariamente ao CPP, não exige que todas as arestas sejam obrigatoriamente servidas. Este problema, exigindo apenas o serviço de um subconjunto de arestas, retrata problemas que melhor representam a realidade.

Introduzido por Golden e Wong (1981), o CARP é um problema em que se pretende determinar rotas que visitam um subconjunto de ligações (arestas) de um determinado grafo. Cada ligação no grafo pode modelar uma rua que pode ser percorrida em qualquer direção e cada nodo representa um cruzamento ou entroncamento. Para além de considerar diferentes conjuntos de veículos para a satisfação de uma determinada procura total, deve ter em consideração as capacidades dos veículos que não podem ser excedidas. Estas, no caso em estudo, representam os horários de trabalho dos AFE.

Segundo Brandão e Eglese (2008), os problemas de roteamento capacitado nos arcos são, regra geral, NP-difíceis. As aplicações do CARP surgem em atividades como a recolha de lixo, a entrega de correio, a limpeza da neve, a limpeza de rua, entre outras. Os autores também consideram que métodos exatos apenas são eficazes em instâncias de

pequena dimensão. Portanto, para resolução de instâncias de maior dimensão recorrem a métodos heurísticos, sendo o primeiro baseado em *Tabu Search* e o segundo num modelo de algoritmos meméticos. A abordagem apresentada por Brandão e Eglese (2008) difere da de outros trabalhos em detalhes que dizem respeito à implementação, em particular, no carácter determinístico do algoritmo.

Labadi et al. (2008) propuseram outro tipo de metaheurística para o CARP, baseado no procedimento *Greedy Randomized Adaptive Search Procedure* (GRASP). Este procedimento começa por gerar uma solução criando um percurso gigante que serve todas as arestas necessárias, ignorando as janelas de tempo e a capacidade dos veículos. Este percurso é depois dividido em percursos admissíveis. De seguida, recorre-se a uma heurística melhorativa de pesquisa local, em que se escolhe o melhor movimento em cada iteração. A heurística melhorativa engloba três movimentos que podem operar em duas rotas no máximo: *Or-opt*, *2-opt* e *Swap*. Além disso, o GRASP proposto é melhorado por um procedimento de *path-relinking*, cuja ideia consiste em explorar um determinado espaço de solução que liga duas soluções de alta qualidade, fornecidas pela heurística melhorativa de pesquisa local.

O *Maximum Benefit Chinese Postman Problem* (MBCPP) foi introduzido por Malandraki e Daskin (1993). O MBCPP é um problema definido em um grafo não orientado, onde cada rota tem início e fim no depósito, sendo maximizado o benefício líquido total. Este é representado como problema de roteamento nos arcos, seja com um único veículo ou vários veículos, onde as arestas a percorrer representam as ruas que podem ser servidas tendo um benefício associado. Segundo os autores, modelar o problema como um MBCPP permite não servir ruas com um baixo benefício e servir várias ruas com um elevado benefício. Ao considerar vários benefícios associados a cada aresta, é possível modelar a situação em que o benefício total depende do número de vezes que as arestas são servidas. Por exemplo, na remoção de neve da rua, a primeira travessia está relacionada a um benefício maior, pois é aquela que geralmente remove a maior parte da neve, enquanto as travessias subsequentes são menos eficazes.

Archetti e Speranza (2014) reclassificam o MBCPP como um *Arc Routing Problem with Profits* (ARPP). Tal como anteriormente o objetivo consiste em determinar quais as ligações a incluir no percurso maximizando o benefício total e tendo em consideração a distância total percorrida. Os veículos possuem uma capacidade limitada

ou existe um tempo limite a respeitar. Os autores distinguem dois tipos diferentes de classes de problemas com base na função objetivo. O *Team Orienteering Arc Routing Problem* (TOARP), cujo objetivo consiste em maximizar o benefício total que respeita uma restrição relativamente à duração do percurso. E o *Prize-Collecting Rural Postman Problem* (PCRPP) que retrata o problema de determinar um percurso que maximize a diferença entre o benefício total obtido e o custo total do percurso.

Archetti et al. (2015) realizaram um estudo sobre o *Team Orienteering Arc Routing Problem* (TOARP). Neste tipo de problemas as potenciais ligações encontram-se representadas por arcos/arestas de um grafo e são escolhidas para adicionar a um percurso com base nos benefícios associados. Um número limitado de veículos encontra-se disponível para satisfazer os clientes. O objetivo consiste em maximizar o benefício total, sendo proposta pelos autores uma *matheuristic* para o TOARP que permitiu obter soluções ótimas para um grande número de instâncias utilizadas (78% daquelas em que a solução ótima era conhecida).

Ferreira (2016) aborda o problema enfrentado pela EMEL, relativamente à otimização dos percursos a realizar pelos fiscais de estacionamento na cidade de Lisboa. O autor descreve uma heurística construtiva e duas heurísticas melhorativas, implementadas utilizando a linguagem de programação VBA. Introduce também grandes conjuntos de dados baseados na rede de estradas do município de Lisboa. O algoritmo desenvolvido no presente trabalho é também testado nesses conjuntos de dados.

Neste capítulo, foi discutido um conjunto de artigos que abordam problemas de roteamento nos arcos, semelhantes quer entre si quer ao proposto neste trabalho. A fiscalização do estacionamento foi definida como um problema de roteamento capacitado com benefícios nos arcos. Além disso, são tidas em conta as diferentes janelas de tempo de exploração, os horários de trabalho e o número limitado de fiscais. O foco em instâncias de grande dimensão exigiu que fossem desenvolvidas abordagens de soluções eficazes para o problema. No capítulo 4, são descritos detalhadamente os métodos heurísticos desenvolvidos para responder ao problema de pesquisa deste projeto.

## **CAPÍTULO IV: METODOLOGIA**

---

### **4.1 Heurística Construtiva Aleatória (HCA)**

#### **4.1.1 Introdução**

Antes de aplicar a Heurística Construtiva Aleatória (HCA) para identificar um conjunto de soluções admissíveis iniciais é necessário definir a hora de início e a duração máxima dos turnos de cada AFE.

Sendo assim, são atribuídos a cada AFE dois percursos diários de trabalho, um por cada turno. No início de cada turno, cada AFE sai do depósito para um ponto base para iniciar o percurso de fiscalização. A escolha do ponto base inicial é feita através de um processo aleatório, e a partir deste, são adicionados arcos ao percurso do AFE, respeitando um conjunto de critérios, enquanto houver tempo disponível no horário de trabalho.

O procedimento iterativo é utilizado para selecionar arcos a adicionar ao percurso. Segue um processo aleatório, optando por segmentos fiscalizáveis e, em último caso, por conectores, uma vez que o objetivo consiste em obter percursos com poucos conectores. Embora a escolha dos arcos seja feita aleatoriamente, pretende-se adicionar ao percurso apenas arcos que ainda possam ser fiscalizados numa determinada hora, sabendo-se que apenas podem ser fiscalizados uma vez em cada duas horas sucessivas. Depois de selecionar o arco, e caso seja possível fiscalizá-lo, o tempo é dado pela soma entre o respetivo tempo de deslocação e o tempo de fiscalização, caso contrário, inclui apenas o tempo de deslocação. De seguida, verifica-se se a inclusão do arco selecionado é compatível com o tempo limite do turno do AFE respetivo. Se o arco for adicionado ao percurso são realizadas as atualizações de variáveis de modo a evitar que este seja novamente fiscalizado durante um período de tempo especificado. Caso não seja compatível, e se não existem mais arcos que possam ser adicionados ao percurso sem violar as restrições do problema, termina-se o percurso corrente com a ligação ao depósito e segue-se para a determinação de um novo percurso para o turno seguinte.

No final, o algoritmo apresenta uma solução admissível inicial como output. Cada solução admissível possui dois percursos diferentes para cada AFE. Estes serão utilizados

como input na metaheurística *Tabu Search*, uma vez que todos os percursos serão sujeitos a pesquisa melhorativa.

#### 4.1.2 Pseudocódigo HCA

A notação necessária para a heurística construtiva aleatória segue a de Ferreira (2016) e consta da Tabela 1.

Notação	Descrição	Definição
$alt_h$	Indica se é necessário incrementar a hora $h$	Inteiro
$cmc_j$	Caminho mais curto entre o nodo $j$ e o depósito passando por um ponto base	Vetor
$crit_{ijh}$	Criticidade do segmento $(i, j)$ na hora $h$	Real
$dt_{k,nt}$	Duração do $nt$ -ésimo turno do AFE $k$	Inteiro
$fisc_{ijh}$	Parâmetro que indica se um determinado arco $(i, j)$ foi fiscalizado na hora $h$	$\{0; h\}$
$fiscalizavel_{ijh}$	$TRUE$ se for possível fiscalizar o arco $(i, j)$ na hora $h$ , e $FALSE$ em c.c.	Booleano
$h$	Hora de exploração/fiscalização	Inteiro
$hi_{k,nt}$	Hora de início do $nt$ -ésimo turno do AFE $k$	Inteiro
$k$	Agente de fiscalização de estacionamento	Inteiro
$ K $	Nº de agentes de fiscalização	Inteiro
$pb_k$	Ponto base usado pelo AFE $k$ para iniciar o percurso	Inteiro
$nt$	Identificação do número de turno	Inteiro
$selecionado$	$TRUE$ se houver pelo menos um arco que possa ser adicionado ao percurso e $FALSE$ em c.c.	Booleano
$soma_t$	Tempo despendido desde a saída do depósito até ao nodo final do último arco adicionado ao percurso	Inteiro
$t_cmc_j$	Tempo de deslocação no $cmc_j$	Real
$tempo_{des_{ij}}$	Tempo de deslocação no arco $(i, j)$	Inteiro
$tempo_{fisc_{ijh}}$	Tempo de fiscalização do arco $(i, j)$ na hora $h$	Inteiro
$tempo_{total}$	Tempo total despendido na deslocação e fiscalização	Inteiro

Tabela 1 – Terminologia utilizada na HCA

De seguida é apresentado, na Figura 5, o procedimento geral da heurística construtiva aleatória.



---

**Pseudocódigo Heurística Construtiva Aleatória (HCA)**

---

**Input:**  $|K|$ ;  $\forall k, \forall nt: hi_{k,nt}; dt_{k,nt}$ ;  $\forall i, j, \forall h: fiscalizavel_{ijh}$

$VS =$  vetor nulo // vetor SA  
 $conta = 0$  // nº de percursos

**Para**  $k = 1$  até  $|K|$   
 $nt = 1$  // nº do turno

**Enquanto**  $nt \leq 2$   
 $VP =$  vetor nulo // vetor percurso  
 $h = hi_{k,nt}$ : *selecionado* = TRUE  
 Escolher aleatoriamente um ponto base,  $pb_k$ , para iniciar o percurso  
 $VP[1] = (dep, pb_k)$ ; Fazer  $i \leftarrow pb_k$ ;  $soma\_t = tempo\_des_{dep, pb_k}$   
 $alt\_h = tempo\_des_{dep, pb_k}$  // se  $alt\_h > 60$ , incrementar  $h$   
 $conta\_num\_arcos = 1$  // nº de arcos no percurso do AFE  
**Enquanto** *selecionado* = TRUE // juntar  $(i, j)$  ao percurso  
     **Se**  $\exists$  segmento  $(i, j)$ :  $fisc_{ijh} = h$  **então**  
         selecionar aleatoriamente um desses segmentos  
     **Senão**  
         selecionar aleatoriamente um conetor com início em  $i$ ,  $(i, j)$   
     **FimSe**  
     **Se**  $fiscalizavel_{ijh} = TRUE$  **então**  
          $tempo\_total = tempo\_des_{ij} + tempo\_fisc_{ijh}$   
     **Senão**  
          $tempo\_total = tempo\_des_{ij}$   
     **FimSe**  
      $tempo\_usado = soma\_t + tempo\_total + t\_cmc_j$   
     **Se**  $tempo\_usado \leq dt_{k,nt}$  **então** // verificação do tempo  
          $conta\_num\_arcos = conta\_num\_arcos + 1$   
          $VP[conta\_num\_arcos] = (i, j)$   
     **Senão** // terminar percurso  
         *selecionado* = FALSE  
          $conta\_num\_arcos = conta\_num\_arcos + 1$   
         Identificar  $pb_k$  mais próximo de  $i$  (incluindo distância para o depósito)  
          $VP[conta\_num\_arcos] = (i, pb_k)$   
          $conta = conta + 1$   
          $VS[conta] = VP$   
         **Se**  $nt = 1$  **então**  $nt = 2$   
         **Senão**  $nt = 1$   
         **FimSe**  
     **FimSe**  
     **Se** *selecionado* = TRUE **então** // atualização de variáveis  
         **Se**  $fiscalizavel_{ijh} = TRUE$  **então**  
              $fisc_{ijh} = h$ ;  $fisc_{ij(h+1)} = h + 1$ ;  $fisc_{ij(h-1)} = h - 1$   
              $fiscalizavel_{ijh} = FALSE$   
              $crit_{ij(h+2)} = 0$ ;  $crit_{ij(h+3)} = 0$   
         **FimSe**  
          $alt\_h = alt\_h + tempo\_total$ ;  $soma\_t = soma\_t + tempo\_total$   
          $i \leftarrow j$   
         **Se**  $alt\_h > 60$  **então**  
              $h = h + 1$ ;  $alt\_h = alt\_h - 60$   
         **FimSe**  
     **FimSe**  
     **FimEnquanto** // selecionado  
     **FimEnquanto** // nt  
      $k = k + 1$

**FimPara**  
**Output**  
 Retornar solução admissível e o respetivo valor da função objetivo

---

**FIM Heurística Construtiva Aleatória**

---

Figura 5 – Pseudocódigo da Heurística Construtiva Aleatória

### 4.1.3 Exemplo de Aplicação da HCA

Sendo útil ilustrar a aplicação das heurísticas propostas, é apresentada uma legenda das figuras relativamente às redes construídas (Figura 6).

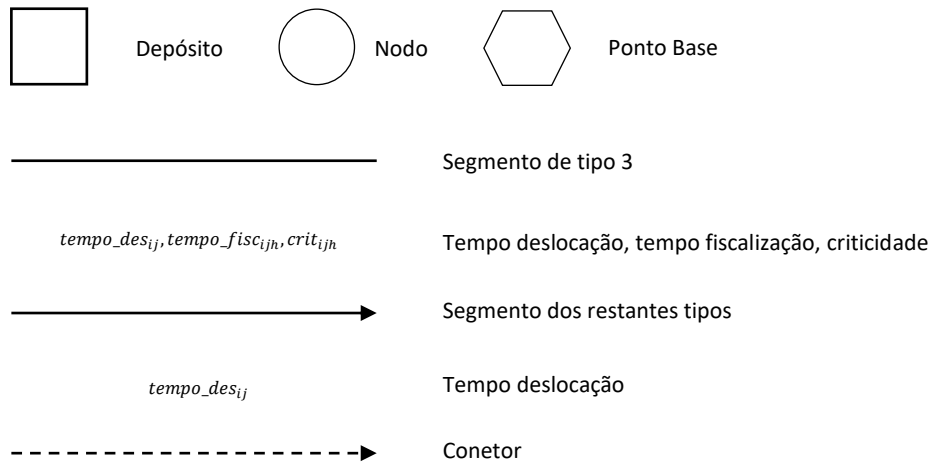


Figura 6 – Legenda para os exemplos

Na figura 7 encontra-se representada uma rede. Esta inclui segmentos e conetores, um depósito (*dep*) correspondente à central da EMEL, três pontos base (nodos 1, 2 e 3) e diversos nodos que servem de referência para a deslocação dos AFE. Os arcos a cheio representam ruas que devem ser fiscalizadas, enquanto as linhas tracejadas representam os conetores. O objetivo consiste em obter dois percursos diários para cada AFE. O turno da manhã inicia às 9 horas e o da tarde às 14, sendo que cada um destes turnos terá a duração máxima de 150 minutos.

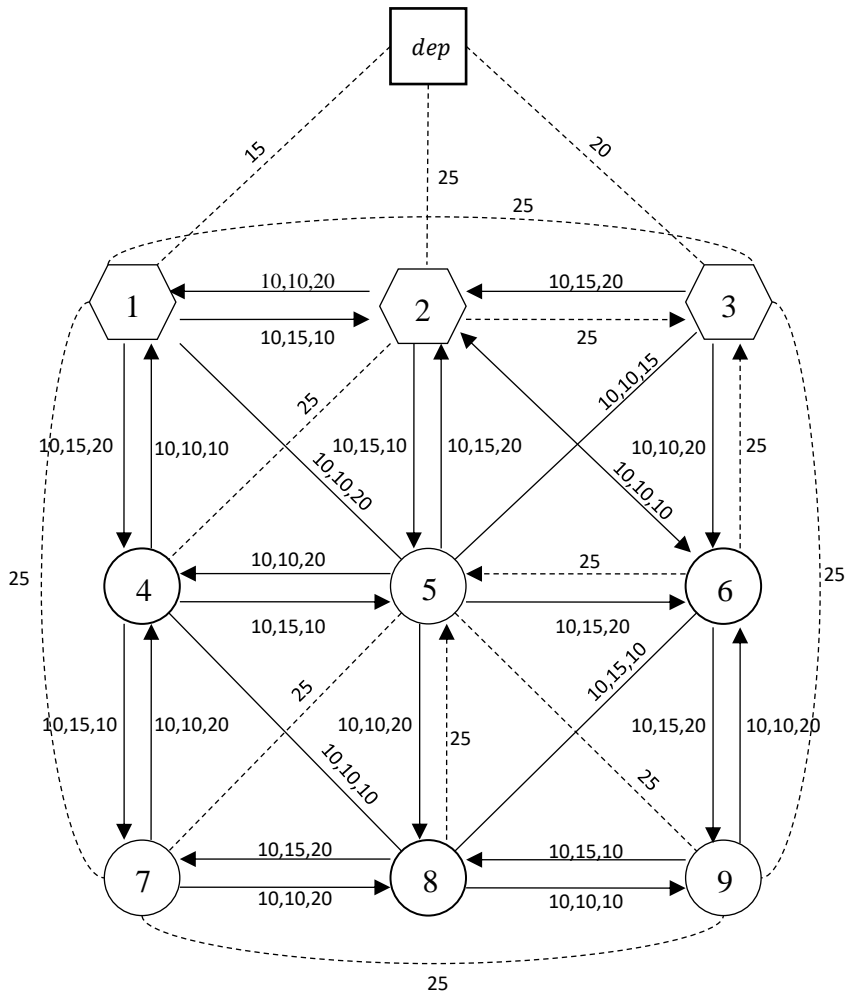


Figura 7 – Rede do exemplo

Na tabela seguinte ilustra-se a aplicação da HCA para a obtenção de uma solução admissível inicial, considerando os dois AFE que realizam dois turnos de trabalho com diferentes percursos. Por questões de simplificação, neste breve exemplo não se assumem alterações no valor das criticidades e no tempo de fiscalização ao longo das diferentes horas de exploração do dia.

<i>k</i>	<i>nt</i>	<i>t<sub>k</sub></i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>tempo_total</i>	<i>t_cmc<sub>j</sub></i>	<i>tempo_usado</i>	<i>soma_t</i>	<i>alt_h</i>	<i>P</i>
1	1	150	9	dep	3				20	20	(dep, 3)
				3	6	20	35	75	40	40	(dep, 3); (3,6)
				6	9	25	45	110	65	65	(dep, 3); (3,6); (6,9)
	10	9	8	25	35	125	90	5+25=30	(dep, 3); (3,6); (6,9); (9,8)		
		8	4	20	25	135	110	50	(dep, 3); (3,6); (6,9); (9,8); (8,4)		
		4	5	25	25	160>150			(dep, 3); (3,6); (6,9); (9,8); (8,4); (4,1); (1, dep)		
4	1,dep				135		0				
	2	150	14	dep	1				15	15	(dep, 1)
				1	4	25	25	65	40	40	(dep, 1); (1,4)
4	5	25	25	90	65	65	(dep, 1); (1,4); (4,5)				
15	5	8	20	35	120	85	5+20=25	(dep, 1); (1,4); (4,5); (5,8)			
	8	6	25	35	145	110	50	(dep, 1); (1,4); (4,5); (5,8); (8,6)			
	6	2	20	25	155>150			(dep, 1); (1,4); (4,5); (5,8); (8,6); (6,2); (6, dep)			
6	2,dep				145		0				
2	1	150	9	dep	1				15	15	(dep, 1)
				1	2	25	25	65	40	40	(dep, 1); (1,2)
				2	5	25	25	90	65	65	(dep, 1); (1,2); (2,5)
	10	5	6	25	35	125	90	5+25=30	(dep, 1); (1,2); (2,5); (5,6)		
		6	2	20	25	135	110	50	(dep, 1); (1,2); (2,5); (5,6); (6,2)		
		2	6	20	35	165>150			(dep, 1); (1,2); (2,5); (5,6); (6,2); (2, dep)		
	2	dep				135		0			
	2	150	14	dep	2				25	25	(dep, 2)
2				6	20	35	80	45	45	(dep, 2); (2,6)	
6	9	25	45	115	70	70	(dep, 2); (2,6); (6,9)				
15	9	8	25	35	130	95	10+25=3	(dep, 2); (2,6); (6,9); (9,8)			
	8	7	25	35	155>150		5	(dep, 2); (2,6); (6,9); (9,8); (8,4); (4,1); (1, dep)			
8	4,1,dep				130						

Legenda: *it* = linha anterior; *tempo\_usado* = *soma\_t* (*it*) + *tempo\_total* + *t\_cmc<sub>j</sub>*; *soma\_t* = *soma\_t* (*it*) + *tempo\_total*

Tabela 2 – Exemplo HCA

O AFE 1 realizará os dois percursos *P1* e *P2* distintos, representados na Figuras 8 e 9, com uma criticidade total de 120 unidades (60 + 60) e uma duração total de 280 minutos (135 + 145).

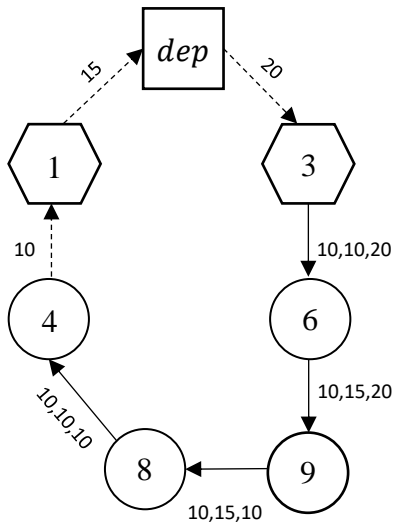


Figura 8 – Primeiro percurso do AFE 1

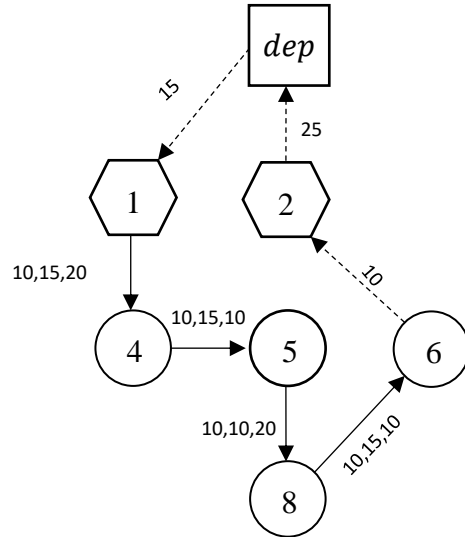


Figura 9 – Segundo percurso do AFE 1

O AFE 2 realiza os dois percursos *P3* e *P4* diferentes, representados nas figuras 10 e 11, respetivamente. Os percursos têm uma criticidade total de 90 unidades (50 + 40) e uma duração total de 265 minutos (135 + 130)

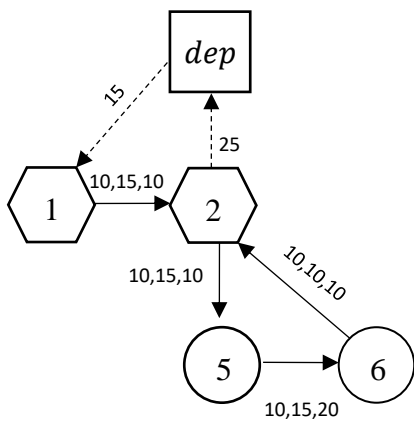


Figura 10 - Primeiro percurso do AFE 2

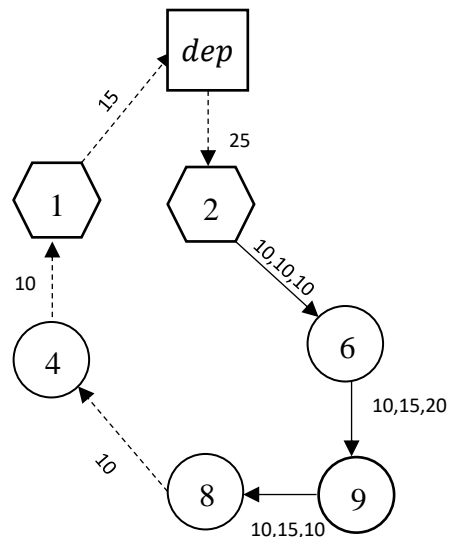


Figura 11 - Segundo percurso do AFE 2

É importante salientar que as soluções admissíveis obtidas com a heurística construtiva aleatória são utilizadas como input para a metaheurística *Tabu Search*. Uma análise detalhada do procedimento metaheurístico desenvolvido é abordada na seção seguinte.

## 4.2 Tabu Search (TS)

### 4.2.1 Apresentação da Metaheurística

O algoritmo *Tabu Search* (TS) é uma metaheurística de pesquisa local, originalmente proposta por Glover (1986), para identificar soluções admissíveis de boa qualidade para problemas complexos de otimização combinatória. O TS tem como propósito tentar ultrapassar o problema enfrentado pelas heurísticas de pesquisa local, que muitas vezes terminam em ótimos locais de fraca qualidade. Para isso, o procedimento admite movimentos para soluções piores e, em cada iteração, um conjunto de informações associadas ao movimento efetuado são armazenadas numa estrutura de memória adaptativa (*tabu*), reduzindo assim o risco de avaliar uma solução mais do que uma vez em iterações sucessivas.

Deste modo, é criada uma lista *tabu* que contém os últimos movimentos realizados até um valor máximo previamente fixado. Um movimento corresponde a um conjunto de novas ligações adicionadas a um percurso que, por sua vez, são declaradas *tabu*. Estas declarações são úteis, pois impedem a pesquisa de retornar de imediato a pontos anteriormente visitados, permitindo assim uma exploração mais extensa da região admissível. Após cada iteração um novo movimento realizado é adicionado à lista, enquanto o mais antigo é removido, caso a dimensão máxima da lista tenha sido atingida. Na exploração da vizinhança da solução corrente ficam excluídas da pesquisa as soluções vizinhas obtidas por movimentos que constam da lista *tabu*.

Entre os conceitos básicos mais importantes do TS surge a definição do espaço de pesquisa e da estrutura de vizinhança. O espaço de pesquisa corresponde ao espaço de todas as soluções admissíveis que podem ser consideradas durante a pesquisa. Por outro lado, em cada iteração entende-se por vizinha de uma solução corrente, uma solução que sofreu uma determinada transformação local. Para o problema em causa, dada uma solução corrente  $S$ , a sua vizinhança é constituída por todas as soluções obtidas de  $S$

através de um movimento que consiste na substituição de segmentos ou conectores em cada percurso atribuído a cada AFE durante o período de fiscalização. A metaheurística TS faz várias iterações a partir de soluções admissíveis iniciais diferentes. O objetivo consiste em melhorar (ou seja, aumentar a criticidade), caso possível, todos os percursos pertencentes a cada solução admissível inicial.

A pesquisa a ser realizada pelo TS pode ser um processo moroso, sendo assim necessário estabelecer critérios de paragem. Neste trabalho foram empregues duas regras como critério de paragem, em que o procedimento é interrompido após atingir o número máximo de iterações: *i*) desde o início da pesquisa; ou *ii*) consecutivas sem verificar qualquer melhoria no valor da função objetivo da melhor solução encontrada até ao momento.

O TS é uma metaheurística que tem vindo a ser aplicada com grande sucesso em vários problemas de otimização.

A notação definida para o *Tabu Search* consta da Tabela 3.

Notação	Descrição	Definição
<b><i>conta_it</i></b>	Contador do número de iterações	Inteiro
<b><i>conta_it_sm</i></b>	Contador do número de iterações sem melhoria em $f^*$	Inteiro
<b><i>dim_T</i></b>	Dimensão da lista <i>tabu</i>	Inteiro
<b><math>f(\bullet)</math></b>	Valor da função objetivo da solução $\bullet$	Real
<b><i>max_dim_T</i></b>	Dimensão máxima da lista <i>tabu</i>	Inteiro
<b><i>max_it</i></b>	Número máximo de iterações	Inteiro
<b><i>max_it_sm</i></b>	Número máximo de iterações sem melhoria em $f^*$	Inteiro
<b><i>mov</i></b>	Arcos referentes ao movimento realizado (adicionados)	Conjunto
<b><math>N(S)</math></b>	Vizinhança de uma solução $S \in X$	$N(S) \subseteq X$
<b><math>S</math></b>	Solução corrente	Solução Admissível ( $\in X$ )
<b><math>S_0</math></b>	Solução inicial	Solução Admissível ( $\in X$ )
<b><math>S^*</math></b>	Melhor solução encontrada até ao momento	Solução Admissível ( $\in X$ )
<b><math>T</math></b>	Lista <i>tabu</i>	Vetor
<b><math>X</math></b>	Conjunto de soluções admissíveis	Conjunto

Tabela 3 – Terminologia utilizada no TS

#### 4.2.2 Descrição Geral do Pseudocódigo TS

O pseudocódigo do TS é apresentado na Figura 12, sendo composto por um passo de inicialização e um iterativo. A inicialização tem como propósito apenas declarar parâmetros essenciais ao processo iterativo. A metaheurística tem como input um conjunto de dados previamente obtidos e, no final, apresenta como output a melhor solução admissível encontrada. Para além disso, é necessário definir valores para os parâmetros que dizem respeito ao número de iterações e à dimensão da lista *tabu*.

Os dados de input para a metaheurística são: *i*) um conjunto de soluções admissíveis obtidas com recurso a uma heurística construtiva; *ii*) número máximo de iterações para a pesquisa, *max\_it*; *iii*) o número máximo de iterações sem verificar qualquer melhoria no valor da função objetivo da melhor solução encontrada até o momento, *max\_it\_sm*; e por fim, *iv*) a dimensão máxima da lista *tabu*, *max\_dim\_T*.

O passo iterativo inclui um procedimento melhorativo, *2-opt*, que gera uma melhor solução vizinha  $S'$ , o valor da respetiva função objetivo,  $f(S')$ , e o movimento realizado, isto é, os novos arcos adicionados ao percurso, *mov*. A estratégia de pesquisa na vizinhança considerada bem como o conceito de vizinhança são apresentados no ponto seguinte. A metaheurística armazena na lista *tabu*, *T*, informação de movimentos realizados recentemente e, de seguida, utiliza-a para evitar movimentos para soluções já visitadas.

Na figura 12 é apresentado, em detalhe, o pseudocódigo da metaheurística *Tabu Search*.



**Pseudocódigo Tabu Search**


---

```

Input
 $X = \{\text{conjunto de soluções admissíveis}\}$  // soluções geradas através da HCA (pág. 12 a 19)
 $max\_it$  // número máximo de iterações
 $max\_it\_sm$  // número máximo de iterações sem melhorar  $f^*$ 
 $max\_dim\_T$  // dimensão máxima da lista tabu
 $f^* = 0$ 
Enquanto  $X \neq \emptyset$ 
  Identificar nova solução admissível  $S_0 \in X$  // em todas as soluções admissíveis de  $X$ 
  Identificar  $N(S_0)$  // vizinhança de uma solução  $S_0 \in X$  (secção 4.3)
   $S = S_0; f(S) = f(S_0)$ 
   $conta\_it = 0$  // contador do número de iterações
  Para  $k = 1$  até  $2 * nAFE$  // nº de percursos
  Enquanto  $conta\_it \leq max\_it$ 
     $conta\_it\_sm = 0$  // contador do número de iterações sem melhoria em  $f^*$ 
     $dim\_T = 0$  // dimensão da lista tabu
     $T = \text{vetor nulo}$  // lista tabu
    Enquanto  $conta\_it\_sm \leq max\_it\_sm$ 
       $conta\_it = conta\_it + 1$ 
      Escolher um  $P$  de  $S$  ainda não selecionado
      Call 2-opt // pseudocódigo na página 25
      Output:  $S', f(S'), mov$  // o TS é aplicado em todos os percursos de cada AFE
      Se  $mov$  não está em  $T$  então
        Se  $f(S') > f^*$  então
           $S^* = S'; S = S'; f^* = f(S'); conta\_it\_sm = 0$ 
        Senão
           $S = S'; conta\_it\_sm = conta\_it\_sm + 1$ 
        FimSe
      Se  $dim\_T = max\_dim\_T$  então // atualização da lista tabu
        Para  $i = 1$  até  $(dim\_T - 1)$ 
           $T(i) = T(i + 1)$ 
        FimPara
           $T(dim\_T) = mov$ 
      Senão
         $dim\_T = dim\_T + 1$ 
         $T(dim\_T) = mov$ 
      FimSe
    FimSe
  FimEnquanto //  $conta\_it\_sm$ 
FimEnquanto //  $conta\_it$ 
  Atualizar  $X$ 
FimPara
FimEnquanto //  $X \neq \emptyset$ 
Output
Retornar  $S^*$  e  $f^*$ 

```

---

**FIM TABU SEARCH**

Figura 12 – Pseudocódigo da metaheurística Tabu Search

O pseudocódigo do *Tabu Search* descrito na Figura 12 inclui um procedimento heurístico melhorativo, *2-opt*. Este procedimento é abordado na secção seguinte.

### 4.3 Heurística *l-opt*

#### 4.3.1 Introdução

Partindo de uma solução admissível inicial, pretende-se alcançar sucessivas melhorias no valor da função objetivo, reconstruindo percursos através da adição e/ou remoção de arcos (segmentos ou conetores).

Um método muito utilizado é a heurística *l-opt*, originalmente proposta por Croes (1958). Dado um percurso  $P$ , representado por um circuito, são removidos  $l$  ou menos arcos não contíguos no percurso, originando  $l$  caminhos separados e, de seguida, são introduzidos outros  $l$  arcos de modo a completar o circuito, dando origem a um novo percurso, designado por  $P'$ . É feita uma exploração na vizinhança e aceita-se um movimento logo que seja encontrada uma solução melhor na vizinhança, seguindo uma estratégia *first-improvement*. Acresce que a pesquisa, neste algoritmo, se torna impraticável à medida que  $l$  aumenta, devido à sua complexidade computacional. Contudo, este pode facilmente encontrar boas soluções para valores baixos de  $l$ .

De seguida, apresenta-se um pequeno exemplo da aplicação desta heurística melhorativa com  $l = 2$ , *2-opt*, valor que será adotado ao longo do trabalho. A Figura 13 representa um exemplo de um percurso inicial,  $P_0 = (1,2,3,4,5,6,7,8,1)$ , enquanto a Figura 14 representa um novo percurso  $P' = (1,2,5,4,3,6,7,8,1)$ . Este exemplo retrata a passagem de uma solução a uma solução vizinha aplicando a heurística *2-opt* em que foram excluídos os arcos  $B = (2,3)$  e  $E = (5,6)$  do percurso inicial e introduzidos os arcos  $I = (2,5)$  e  $J = (3,6)$ . No entanto, verifica-se que, de modo a completar novamente o percurso, foi necessário inverter também o sentido dos arcos no caminho entre 3 (nodo final do primeiro arco removido) e 5 (nodo inicial do segundo arco removido). Ou seja, os arcos  $C = (3,4)$  e  $D = (4,5)$ , foram invertidos resultando nos arcos do novo percurso  $C' = (4,3)$  e  $D' = (5,4)$ , respetivamente. A solução que resulta da inicial,  $S$ , trocando percursos da solução inicial,  $P$ , por novos percursos,  $P'$ , considera-se vizinha de  $S$ .

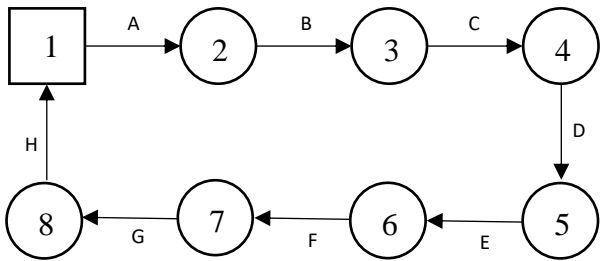


Figura 13 - Percurso  $P_0$  na solução corrente

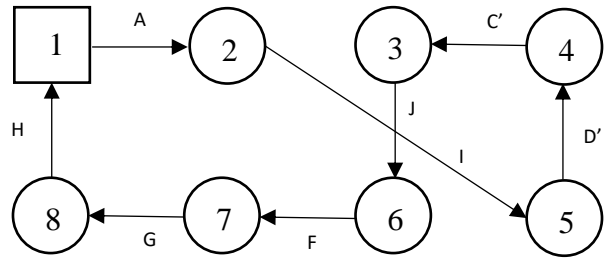


Figura 14 - Percurso  $P'$  obtido pela heurística 2-opt

### 4.3.2 Descrição do Pseudocódigo Heurística 2-opt

O procedimento geral da heurística 2-opt é composto por um passo de inicialização e por um iterativo.

A inicialização do procedimento identifica: *i*) o conjunto  $ap$ , constituído por todos os arcos que formam o percurso inicial; *ii*) se existe caminho,  $CMC$ , entre os diferentes pares de nodos no percurso.

No procedimento iterativo são escolhidos os arcos a eliminar e os arcos a incluir no percurso em análise, bem como o caminho a inverter de modo a completar o percurso. Este é repetido enquanto existirem arcos que possam ser retirados do percurso inicial ou se não for encontrado um movimento possível. Escolhem-se aleatoriamente dois nodos,  $j$  e  $k$ , ( $j \neq k$ ) para os quais existe um caminho não singular no percurso  $CMC(j, k)$  e que são extremidades de arcos não adjacentes nesse percurso. Os arcos  $(i, k)$  e  $(j, l)$  são adicionados ao percurso, em substituição de  $(i, j)$  e  $(k, l)$ , e é invertido o sentido dos arcos em  $CMC(j, k)$ , reconstruindo assim um novo percurso que, por sua vez, dará origem a uma nova solução admissível  $S'$ . Se o valor da função objetivo da nova solução for superior ao da melhor solução encontrada até ao momento, faz-se a atualização das variáveis e do valor da função objetivo. Os novos arcos adicionados ao percurso são registados como um movimento,  $mov$ . Sendo assim, neste algoritmo a pesquisa termina quando um ótimo local é encontrado.

No final da heurística 2-opt obtém-se como output uma solução admissível vizinha e a respetiva transformação local efetuada.

De seguida é apresentado, na Figura 15, o pseudocódigo da Heurística 2-opt.

---

### Pseudocódigo 2-OPT

---

**Input**

$S; S^*; f^*$   
 $P$  // Percurso  $P$  entra como input  
 $ap$  // conjunto de arcos pertencentes ao percurso  $P$  inicial  
 $CMC$  //caminho mais curto entre dois quaisquer nodos de  $P$  em n° de arcos  
 $ca = ca \cup_{\{(j,k)\}} \{(j,k): (i,j), (k,l) \in ap \wedge (j,k) \notin ap \wedge 1 < CMC(j,k) < \infty\}$  // arcos entre possíveis pontos de troca  
 $P' = P$   
 $mov = \emptyset$   
 $S' = S$

**Enquanto**  $ca \neq \emptyset$  ou  $mov \neq \emptyset$ 

$S' = S \setminus P$   
 Escolher aleatoriamente  $(j,k) \in ca$   
 $P' = P \setminus \{(i,j), (k,l) \in ap: (j,k) \in ca\}$  // remover um par de arcos do percurso inicial  
 $P' = P' \cup \{(i,k), (j,l)\}$  // adicionar dois novos arcos  
 Inverter em  $P'$  todos os arcos em  $CMC(j,k)$   
 $S' = S' \cup P'$  // identificar o novo  $S'$  trocando  $P$  por  $P'$   
 Se  $f(S') > f^*$  então  
 $mov = \{(i,k), (j,l)\}$   
**FimSe**  
 $ca = ca \setminus \{(j,k)\}$

**FimEnquanto****Output**

Retornar  $S', f(S'), mov$

---

**FIM 2-OPT**

Figura 15 – Pseudocódigo da Heurística 2-opt

A HCA permite gerar soluções admissíveis rapidamente, no entanto, a qualidade das soluções é relativamente fraca. Por outro lado, com maior tempo de execução a metaheurística *Tabu Search* permite melhorar as soluções iniciais, aumentando o valor da criticidade total de um percurso, caso seja possível. O capítulo 5 analisa os resultados obtidos quando os procedimentos desenvolvidos são executados.

## CAPÍTULO V: ANÁLISE COMPARATIVA DE RESULTADOS

Ao longo deste capítulo apresentam-se os resultados obtidos em dois testes, bem como uma análise da metodologia desenvolvida. Para avaliar a eficácia e viabilidade dos algoritmos propostos, e o impacto da sua aplicação, foram conduzidas várias experiências computacionais. Os programas foram implementados em VBA EXCEL, e nos testes foi usada uma instância com base na rede rodoviária do município de Lisboa. Os dados relativos à instância incluem duas zonas (zonas 1 e 2), que podem ser observadas no Anexo 1. Este conjunto de zonas possui, na totalidade, 38 nodos e 59 ligações. Em cada teste foram obtidas cinco soluções admissíveis, cada uma com dois percursos distintos para cada AFE. Os valores relevantes para as soluções admissíveis podem ser observadas nos anexos.

Neste capítulo, é efetuado um estudo sobre o desempenho da metaheurística *Tabu Search*. São efetuados dois testes em que se consideram diferentes valores para os critérios de paragem. O objetivo consiste na identificação dos principais fatores do algoritmo que podem melhorar ou piorar o seu desempenho. Os dados relativamente aos valores atribuídos aos parâmetros da metaheurística em cada um dos testes podem ser observados na Tabela 4.

	Teste 1	Teste 2
<i>max_it</i>	10	20
<i>max_it_sm</i>	5	10
<i>max_dim_T</i>	4	4
<i>Nº de AFE</i>	2	2

Tabela 4 – Valores para os parâmetros

### 5.1 Teste Computacional 1

A Tabela 5 apresenta os resultados médios obtidos para um conjunto de cinco soluções admissíveis. Verifica-se que a metaheurística *Tabu Search* supera, em média, a heurística construtiva em termos de qualidade de solução. Relativamente ao tempo computacional, obviamente que a metaheurística exige maior tempo, face ao da heurística construtiva, uma vez que faz uma pesquisa mais exaustiva na vizinhança da solução com o objetivo de a melhorar. Para o AFE 1 observaram-se melhorias ligeiras no valor médio da função objetivo, contudo, para o AFE 2 foi possível alcançar uma melhoria significativa de 67% quando aplicada a metaheurística.

Por outro lado, é importante realçar que a média da percentagem de tempo em fiscalização das soluções geradas pela heurística construtiva para o AFE 1 e para o AFE 2 são de 57,4% e 41,9%, respetivamente. Enquanto nas soluções da metaheurística a média da percentagem do tempo em fiscalização aumentou ligeiramente para o AFE 1 (57,9%), mas significativamente para o AFE 2 (56,6%). Relativamente à SA global, verificou-se um aumento relativo da criticidade média de 29,9%, o que permite afirmar que a metaheurística TS é capaz de gerar melhores soluções que a HCA. Estes resultados indicam que a metaheurística também adiciona muitos conectores ao percurso, embora substitua segmentos com baixa criticidade por segmentos com maior criticidade.

	Método	Criticidade média	Aumento Criticidade média		Duração média (min)	Média da % do tempo em fiscalização	Tempo computacional médio (seg)
			Absoluto	Relativo			
AFE 1	HCA	7721,9	-	-	185	57,4	1,9
	TS	7989	267,1	3,5	189	57,9	23,1
AFE 2	HCA	5510,6	-	-	193	41,9	2,8
	TS	9205,2	3694,6	67	192,8	56,6	24
SA	HCA	13232,5	-	-	378	49,5	4,7
	TS	17194,2	3961,7	29,9	381,8	57,3	47,1

Tabela 5 - Resultados médios para o teste 1

## 5.2 Teste Computacional 2

Para a segunda experiência computacional segue-se o mesmo procedimento, embora agora com diferentes valores para os critérios de paragem. Neste caso obtiveram-se tempos de processamento maiores devido ao aumento dos valores associados aos parâmetros da metaheurística. Na Tabela 6, são apresentados os resultados médios obtidos para o teste computacional 2.

As soluções iniciais geradas apresentaram uma qualidade razoável. Novamente, houve aumentos ligeiros da criticidade média das soluções geradas para o primeiro fiscal, mas superiores aos obtidos no primeiro teste. Com a aplicação do TS houve um aumento de 46,3% na criticidade média para o AFE 2, constituindo um resultado bastante positivo. Relativamente à média de percentagem do tempo em fiscalização nas soluções iniciais e nas melhoradas observou-se novamente aumentos quando a metaheurística foi aplicada. No teste 2, para a SA global obteve-se um aumento relativo da criticidade média de

26,3%. Embora se esperasse um maior aumento, uma vez que os valores dos parâmetros estabelecidos para este teste eram superiores aos do teste anterior, este resultado é considerado satisfatório. Em suma, os resultados aparentam ser semelhantes aos do teste anterior, apenas com ligeiras diminuições na percentagem do tempo de fiscalização para o AFE 1.

AFE	Método	Criticidade média	Aumento Criticidade média		Duração média (min)	Média da % do tempo em fiscalização	Tempo computacional médio (seg)
			Absoluto	Relativo			
AFE 1	HCA	7374,8	-	-	180,1	53,3	2
	TS	7855,7	480,9	6,5	183,1	56	45,4
AFE 2	HCA	7257,1	-	-	190,9	49,6	2,4
	TS	10617,1	3360	46,3	185,4	60,3	43,4
SA	HCA	14631,9	-	-	371	51,4	4,4
	TS	18472,8	3840,9	26,3	368,5	58,2	88,8

Tabela 6 - Resultados médios para o teste 2

Os resultados obtidos com a heurística construtiva, por vezes, superaram os esperados gerando soluções adequadas de forma aleatória e com tempos de computação relativamente curtos. De um modo geral, os resultados apontam o *Tabu Search* como sendo capaz de gerar soluções de boa qualidade para o problema de roteamento capacitado nos arcos, com um tempo de processamento baixo. A comparação do valor da criticidade média das soluções para os testes 1 e 2 podem ser vistos nas Figuras 16 e 17, respetivamente.

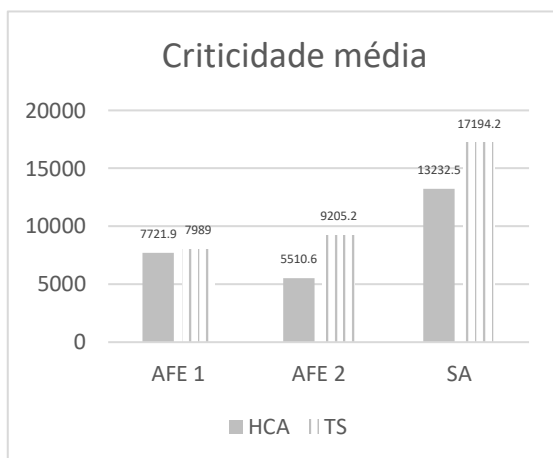


Figura 16 - Criticidade média teste 1

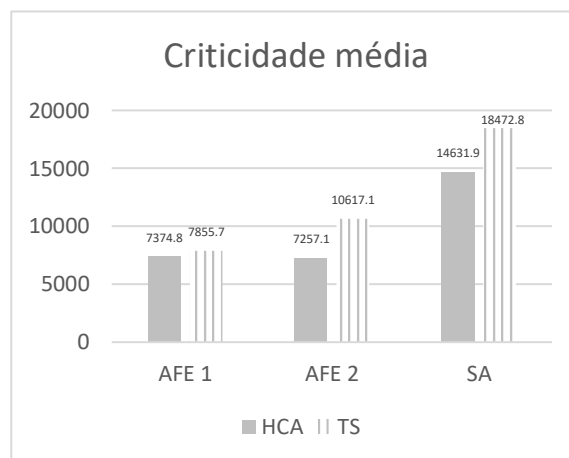


Figura 17 - Criticidade média teste 2

## CAPÍTULO VI: CONCLUSÕES

---

Atualmente, observa-se que o estacionamento afeta a sociedade em geral, desde o tráfego, aos residentes de uma rua e também aos visitantes de curto prazo. A EMEL, enquanto empresa que faz a gestão do estacionamento na cidade de Lisboa, apercebeu-se que os métodos de monitoramento atuais se revelaram ineficientes, suscitando assim a necessidade de melhorias. Este projeto evidencia que a aplicação de heurísticas desencadeia uma série de benefícios na fiscalização do estacionamento.

Neste trabalho final de mestrado foi proposta uma heurística construtiva e uma abordagem metaheurística de tipo *Tabu Search* que inclui um procedimento *2-opt* para a resolução de um problema de roteamento capacitado nos arcos com benefícios e incluindo vários veículos (fiscais). Para verificar o desempenho dos algoritmos propostos, foram conduzidos testes computacionais em duas instâncias. A heurística construtiva permitiu obter soluções em pouco tempo computacional. Por outro lado, a metaheurística revelou-se importante na melhoria das soluções obtidas pela construtiva, gerando, em geral, soluções com maior valor da função objetivo, exigindo um tempo de computação que embora seja maior, não é muito elevado. Após vários testes verificou-se que os movimentos *tabu*, definidos na implementação do *Tabu Search* constituíram um dos fatores mais importantes pois, para além de evitarem pesquisas em soluções previamente visitadas, conseguiram melhorar as soluções iniciais a um ritmo relativamente rápido.

Além da metaheurística *Tabu Search*, que foi alvo de investigação neste projeto, vários outros algoritmos heurísticos poderiam também ser utilizados para gerar soluções de melhor qualidade, nomeadamente: GRASP, *Simulated Annealing*, algoritmo Genético, entre outros. Na literatura, existe uma grande variedade de heurísticas para o problema de roteamento capacitado nos arcos, embora poucas abordem a questão da fiscalização do estacionamento, em particular. O desempenho de uma metaheurística depende dos valores atribuídos aos seus parâmetros, normalmente os relacionados com o critério de paragem, evidenciando um ponto importante nestes algoritmos. Realizar uma investigação para comparar os resultados obtidos por diferentes metaheurísticas e o tempo necessário para gerar soluções poderia fornecer uma melhor ideia sobre qual permite obter as melhores soluções.



## REFERÊNCIAS BIBLIOGRÁFICAS

---

- Amberg, A., Domschke, W., & Voß, S. (2000). Multiple Center Capacitated Arc Routing Problems: A Tabu Search Algorithm using Capacitated Trees. *European Journal of Operational Research*, 124, 360-376.
- Archetti, C., & Speranza, M. G. (2014). Arc Routing Problems with Profits. Em Á. Corberán, & G. Laporte, *Arc Routing: Problems, Methods, and Applications*. Philadelphia: MOS-SIAM Series on Optimization.
- Archetti, C., Corberán, Á., Plana, I., Sanchis, J. M., & Speranza, M. G. (2015). A Metaheuristic for the Team Orienteering Arc Routing Problem. *European Journal of Operational Research*, 245, 392-401.
- Armas, J., Keenan, P., Juan, A., & McGarraghy, S. (2019). Solving Large-Scale Time Capacitated Arc Routing Problems: From Real-Time Heuristics to Metaheuristics. *Annals of Operations Research*, 273, 135-162.
- Beullens, P., Muyldermans, L., Cattrysse, D., & Oudheusden, D. (2003). A Guided Local Search Heuristic for the Capacitated Arc Routing Problem. *European Journal of Operational Research*, 147, 629-643.
- Brandão, J., & Eglese, R. (2008). A Deterministic Tabu Search Algorithm for the Capacitated Arc Routing Problem. *Computers & Operations Research*, 35, 1112-1126.
- Chen, Y., Glover, F., & Hao, J. (2016). A Hybrid Metaheuristic Approach for the Capacitated Arc Routing Problem. *European Journal of Operational Research*, 253, 25-39.
- Corberán, Á., & Laporte, G. (2014). *Arc Routing Problems, Methods, and Applications*. Philadelphia: MOS-SIAM Series on Optimization.
- Corberán, Á., Martí, R., & Romero, A. (2000). Heuristics for the Mixed Rural Postman Problem. *Computers & Operations Research*, 27, 183-203.
- Cortinhal, M., Mourão, M., & Nunes, A. (2016). Local Search Heuristics for Sectoring Routing in a Household Waste Collection Context. *European Journal of Operational Research*, 255, 68-79.
- Croes, G. (1958). A Method for Solving Travelling-Salesman Problems. *Operations Research*, 6, 791-812.
- EMEL. (2017). *Plano de Atividades e Orçamento para 2017*. Obtido em Setembro de 2019, de EMEL - Empresa de Mobilidade e Estacionamento de Lisboa: [https://www.emel.pt/fotos/financeiro/emel-pao\\_2017-completo\\_16432151595984b41527f5c.pdf](https://www.emel.pt/fotos/financeiro/emel-pao_2017-completo_16432151595984b41527f5c.pdf)
- Ferreira, R. (2016). *Otimização de Percursos de Fiscalização Apeada de Estacionamento*. Lisboa: Trabalho Final de Mestrado em Métodos Quantitativos

para a Decisão Económica e Empresarial, Universidade de Lisboa, Instituto Superior de Economia e Gestão.

Ghiani, G., Guerriero, F., Laporte, G., & Musmanno, R. (2004). Tabu Search Heuristics for the Arc Routing Problem with Intermediate Facilities under Capacity and Length Restrictions. *Journal of Mathematical Modelling and Algorithms*, 3, 209-223.

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*, 13, 533-549.

Golden, B. L., & Wong, R. T. (1981). Capacitated Arc Routing Problems. *Networks*, 11, 305-315.

Guan, M. (1962). Graphic Programming using Odd and Even Points. *Chinese Mathematics* 1, 273-277.

Malandraki, C., & Daskin, M. (1993). The Maximum Benefit Chinese Postman Problem and The Maximum Benefit Traveling Salesman Problem. *European Journal of Operational Research*, 65, 218-234.

Mansfield, R. (2008). *Mastering VBA for Office 2007, 2nd Edition*. Sybex.

Mourão, M. C., & Pinto, L. S. (2017). An Updated Annotated Bibliography on Arc Routing Problems. *Networks*, 70(3), 144-194.

Sureja, N., & Chawda, B. (2012). Random travelling Salesman Problem using SA. *International Journal of Emerging Technology and Advanced Engineering*, 2, 2250-2459.

Usberti, F., França, P., & França, A. (2013). GRASP with Evolutionary Path-Relinking for the Capacitated Arc Routing Problem. *Computers & Operations Research*, 40, 3206-3217.

## ANEXOS

### Anexo 1 – Rede

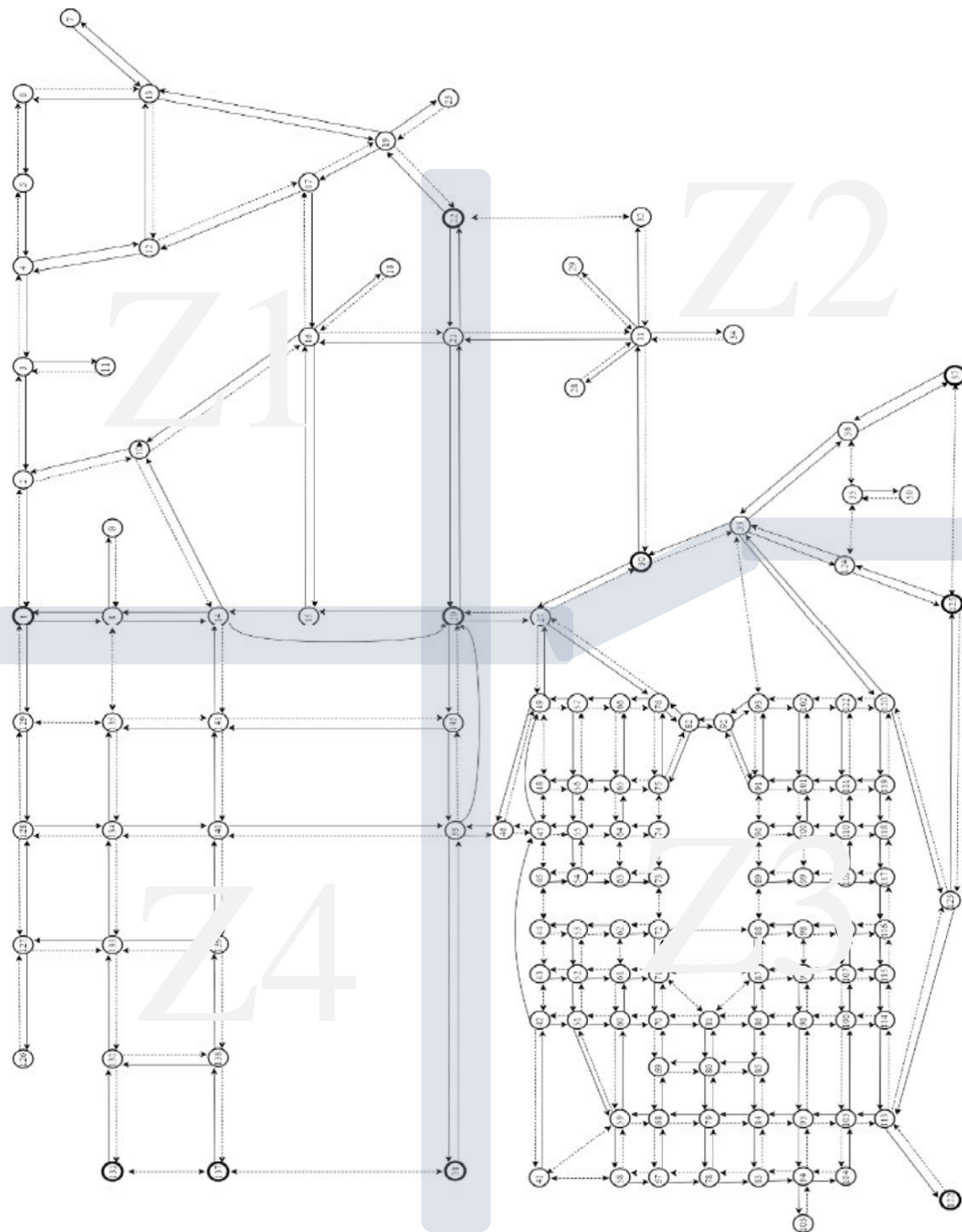


Figura 18 – Rede com quatro zonas de fiscalização (Fonte: Ferreira, 2016)

**Anexo 2 – Soluções Admissíveis (Teste 1)**

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	7955	-	-	178	51,7	1,9
		TS	7955	0	0	178	51,7	23,9
	Tarde	HCA	12922	-	-	181	63,5	2,1
		TS	13274	352	2,7	187	63,6	22,9
2	Manhã	HCA	4128	-	-	194	52,6	2,9
		TS	6509	2381	57,7	198	57,1	31,5
	Tarde	HCA	5108	-	-	197	61,9	1,6
		TS	5108	0	0	197	61,9	16,3

Tabela 7 – Resultados solução admissível 1 (T1)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	3255	-	-	176	61,4	1,6
		TS	3255	0	0	176	61,4	17
	Tarde	HCA	5708	-	-	198	62,6	2,2
		TS	5708	0	0	198	62,6	25,1
2	Manhã	HCA	6710	-	-	197	48,2	2,9
		TS	9356	2646	39,4	198	64,1	24,5
	Tarde	HCA	3720	-	-	198	30,8	3,5
		TS	7259	3539	95,1	199	47,2	29,2

Tabela 8 - Resultados da solução admissível 2 (T1)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	9985	-	-	197	59,4	2,3
		TS	9985	0	0	197	59,4	25,4
	Tarde	HCA	4749	-	-	190	54,2	2,1
		TS	4749	0	0	190	54,2	24,9
2	Manhã	HCA	12975	-	-	193	29,5	3,7
		TS	18630	5655	43,6	198	51,5	27,2
	Tarde	HCA	3959	-	-	183	30,6	2,8
		TS	13479	9520	240,5	194	48,5	25,1

Tabela 9 - Resultados da solução admissível 3 (T1)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	3322	-	-	185	50,8	2
		TS	3539	217	6,5	191	52,4	25
	Tarde	HCA	11304	-	-	164	51,2	1,5
		TS	12213	909	8	177	54,8	22,1
2	Manhã	HCA	4029	-	-	197	56,9	1,8
		TS	4029	0	0	197	56,9	21,8
	Tarde	HCA	4992	-	-	198	48,5	2,8
		TS	5559	567	11,4	176	69,9	17

Tabela 10 - Resultados da solução admissível 4 (T1)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	4526	-	-	186	45,7	2,5
		TS	5140	614	13,6	198	49	27,2
	Tarde	HCA	13493	-	-	195	73,3	1,1
		TS	14072	579	4,3	198	70,7	17,3
2	Manhã	HCA	2893	-	-	177	35	
		TS	3590	697	24,1	175	52,6	24,7
	Tarde	HCA	6592	-	-	196	25	
		TS	18533	11941	181,1	196	56,6	23

Tabela 11 - Resultados para a solução admissível 5 (T1)

### Anexo 3 – Soluções Admissíveis (Teste 2)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	4595	-	-	195	57,4	1,7
		TS	4595	0	0	195	57,4	45,3
	Tarde	HCA	6338	-	-	175	49,7	1,53
		TS	8533	2195	34,6	198	51,5	46,3
2	Manhã	HCA	8736	-	-	196	70,4	1,5
		TS	8736	0	0	196	70,4	35,1
	Tarde	HCA	6913	-	-	191	47,6	2,5
		TS	12907	5994	86,7	194	66	43,2

Tabela 12 - Resultados para a solução admissível 1 (T2)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	7650	-	-	153	50,9	1,8
		TS	7650	0	0	153	50,9	44,5
	Tarde	HCA	8285	-	-	163	66,8	1,5
		TS	10247	1962	23,7	173	64,2	40,3
2	Manhã	HCA	7615	-	-	199	44,2	3,1
		TS	10754	3139	41,2	186	55,4	54,7
	Tarde	HCA	8442	-	-	199	46,7	2,6
		TS	9089	647	7,7	198	48,5	50,6

Tabela 13 - Resultados para a solução admissível 2 (T2)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	7613	-	-	167	59,9	1,5
		TS	7613	0	0	167	59,9	36,4
	Tarde	HCA	4344	-	-	188	33,5	3,5
		TS	4996	652	15	185	56,2	46,2
2	Manhã	HCA	2050	-	-	189	29,1	3,1
		TS	13845	11795	575,4	187	62,6	39,7
	Tarde	HCA	9344	-	-	161	53,4	1,71
		TS	9344	0	0	161	53,4	43,6

Tabela 14 - Resultados para a solução admissível 3 (T2)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	3254	-	-	196	50	2,9
		TS	3254	0	0	196	50	63,1
	Tarde	HCA	9498	-	-	191	50,8	2,1
		TS	9498	0	0	191	50,8	43,6
2	Manhã	HCA	4324	-	-	183	50,3	1,8
		TS	6336	2012	46,5	181	60,2	34,7
	Tarde	HCA	8700	-	-	197	42,6	2,7
		TS	11548	2848	32,7	183	60,7	42,9

Tabela 15 - Resultados para a solução admissível 4 (T2)

AFE	Turno	Método	Críticidade	Aumento Críticidade		Duração (min)	% de tempo em fiscalização	Tempo computacional (seg)
				Absoluto	Relativo			
1	Manhã	HCA	7163	-	-	180	50	2,2
		TS	7163	0	0	180	50	52,8
	Tarde	HCA	15008	-	-	193	64,2	1,6
		TS	15008	0	0	193	64,2	35,1
2	Manhã	HCA	8255	-	-	198	52,5	2,4
		TS	9501	1246	15,1	185	64,3	42,9
	Tarde	HCA	8192	-	-	196	58,7	2,4
		TS	14111	5919	72,3	183	61,7	46,6

Tabela 16 - Resultados para a solução admissível 5 (T2)