

**Universidade do Minho**  
Escola de Engenharia

João Afonso Vieira Casal

*Vehicle Routing Problems*  
**Investigação e construção de um  
Sistema de Informação Geográfica**



**Universidade do Minho**

Escola de Engenharia

João Afonso Vieira Casal

*Vehicle Routing Problems*

**Investigação e construção de um  
Sistema de Informação Geográfica**

Dissertação de Mestrado  
Mestrado em Engenharia de Sistemas

Trabalho realizado sob a orientação do  
**Doutor José António Vasconcelos Oliveira**  
e do  
**Doutor Pedro Rangel Henriques**

Outubro de 2012

## DECLARAÇÃO

Nome

João Afonso Vieira Casal

Endereço electrónico: joaocasal@gmail.com Telefone: 914739701

Número do Bilhete de Identidade: 11586982

Título dissertação /tese

Vehicle Routing Problem - Investigação e construção de um Sistema de Informação Geográfica

\_\_\_\_\_

Orientador(es):

Professor Doutor José António Oliveira e Professor Doutor Pedro Rangel Henriques

\_\_\_\_\_ Ano de conclusão: 2012

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento:

Mestrado em Engenharia de Sistemas

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 31 / 10 / 2012

Assinatura: \_\_\_\_\_

## Agradecimentos

---

A presente dissertação é o fruto de muitas horas de esforço e dedicação, e representa o investimento feito para a consecução de um objetivo. Porém, tal apenas foi conseguido com o apoio e força dados por várias pessoas, nomeadamente:

A minha esposa Sílvia, que sempre compreendeu sem questionar as horas em que o trabalho me afastou, tornando-me a vida além da dissertação mais leve em tudo o que pôde.

O Professor Doutor José António Oliveira que utilizando alguma psicologia de motivação me manteve focado e de olhos sobre o objetivo, direcionando-me na investigação da área da dissertação e levantando problemáticas de reflexão que conduziram a um trabalho mais completo e sólido.

O Professor Doutor Pedro Rangel Henriques cujo apoio muito aprecio desde os tempos da minha licenciatura por ser um profissional que admiro, pelo seu tacto e serenidade na interação e pela experiência que lhe permite dar um auxílio fulcral em termos de concepção e análise de qualquer ferramenta informática ou documento científico.

Aos meus pais e irmãs, cujo apoio incondicional me manteve sempre seguro e me impulsionou para avançar.

A todos os que apesar de não serem mencionados contribuíram direta ou indiretamente para a consecução desta dissertação.

*Agradeço sinceramente a todas as pessoas acima mencionadas, sem as quais este trabalho não teria sido tão conseguido.*





## Resumo

---

O problema de encaminhamento de veículos ou *Vehicle Routing Problem* (VRP), apesar de ser estudado há mais de 5 décadas, toma nos dias de hoje especial relevo devido à escalada de custos associados ao sector da logística, que acompanham a subida do preço dos combustíveis. Paralelamente a este fator surgem avanços tecnológicos baseados na Web que permitem a obtenção, representação e análise de dados georreferenciados e textuais em aplicações próprias.

Esta dissertação propõe-se a combinar estudos relativos ao problema de encaminhamento de veículos com tecnologias informáticas de cartografia digital baseadas na Web, para chegar a uma ferramenta de apoio à decisão na referida área da logística.

No presente documento apresentar-se-á inicialmente o resultado de uma investigação sobre os VRP, seus constituintes, objetivos genéricos e principais instâncias, avançando-se de seguida para um estudo sobre as formas de resolução dos mesmos. Nesse ponto serão apresentados algoritmos exatos, heurísticas, meta-heurísticas e aproximações multiobjectivos que são bibliograficamente propostas para resolver a problemática.

Apresentar-se-á de seguida o desenvolvimento de um Sistema de Informação Geográfica (SIG) de apoio à decisão. Este permitirá ao decisor albergar informação textual sobre o seu VRP, sendo que a informação cartográfica será construída com base nas informações prestadas e em dados obtidos automaticamente em serviços Web da aplicação *Google Maps*. A ferramenta facultará ao utilizador propostas de solução para o problema utilizando duas heurísticas construtivas e um método de melhoria local. Em nome da flexibilidade do sistema também será possível ao decisor gerar as suas rotas sem os referidos automatismos. Para demonstrar as potencialidades do sistema apresentar-se-ão vários testes à aplicação desenvolvida.

Acredita-se que a ferramenta em questão além de permitir a análise e compreensão detalhada das heurísticas utilizadas devido à representação gráfica dos resultados das mesmas, tem aplicabilidade em casos reais e potencial de crescimento e adaptação que lhe permitem ser a base de outras ferramentas construídas para casos específicos.

Palavras-chave: Problema de Encaminhamento de Veículos, Logística, Otimização de Rotas, Sistemas de Informação Geográfica, Suporte à Decisão, Redução de Custos de Combustível

# Abstract

---

Despite being studied for more than 5 decades, the Vehicle Routing Problems (VRP) takes on special importance nowadays due to the escalating costs associated with logistics that come along with the rise of the fuel price. Alongside with this factor, there are technological advances emerging that enable Web-based retrieval, representation and analysis of spatial and textual data.

The purpose of this dissertation is to combine an investigation about the problem of vehicle routing with computer technologies of Web-based digital mapping, to reach a decision support tool in that logistics area.

In this document, the results of an investigation into the VRP will be presented, namely its components, objectives and generic instances, following up to a study about the methods to solve them. At this point will be studied exact algorithms, heuristics, meta-heuristics and multi-purpose approximations that are bibliographically proposed to solve the problem.

Subsequently will be presented the development of a Geographic Information System (GIS) for decision support where the decision maker may accommodate textual information about his VRP. The cartographic data will be built based on the information provided by the user and on data that will be automatically provided by Google Maps Web services. The application will provide to the user proposals for solving the problem using two constructive heuristics and a method of local improvement. On behalf of the flexibility of the system, it will also be possible to the decision maker to generate his own routes without the automation implicit on referred heuristics. To demonstrate the capabilities of the software, various tests will be presented.

It is believed that the developed tool, in addition to offer detailed analysis and understanding of the heuristics used due to the graphical representation of the results, has applicability to real cases and potential of growth and adaptation that allow it to be the basis of other tools built for specific cases.

Keywords: Vehicle Routing Problems, Logistics, Route Optimization, Geographic Information Systems, Decision Support, Reducing Fuel Costs

# Índice

---

Agradecimentos .....	iii
Resumo .....	v
Abstract .....	vi
Lista de siglas/abreviaturas.....	xi
Índice de Figuras.....	xiii
1. Introdução.....	1
1.1. Motivação.....	1
1.2. Objetivo .....	1
1.3. Visão Global.....	2
1.4. Estrutura da dissertação .....	2
2. <i>Vehicle Routing Problem</i> .....	5
2.1. Enquadramento.....	5
2.2. Conceitos básicos.....	7
2.2.1. Rede rodoviária .....	8
2.2.2. Clientes .....	8
2.2.3. Depósitos.....	8
2.2.4. Veículos .....	9
2.2.5. Condutores .....	9
2.2.6. Rotas .....	9
2.3. Objetivos .....	10
2.4. Instâncias.....	10
2.4.1. <i>Capacitated VRP (CVRP)</i> .....	11

2.4.2.	VRP <i>with Time Windows</i> (VRPTW) .....	11
2.4.3.	VRP <i>with Backhauls</i> (VRPB) .....	11
2.4.4.	<i>Multiple Depot</i> VRP (MDVRP) .....	12
2.4.5.	VRP <i>with Pick-up and Delivery</i> (VRPPD) .....	12
2.4.6.	<i>Split Delivery</i> VRP (SDVRP) .....	12
2.4.7.	<i>Stochastic</i> VRP (SVRP) .....	13
2.4.8.	<i>Periodic</i> VRP (PVRP) .....	13
3.	Algoritmos para resolução de VRP .....	15
3.1.	Algoritmos exatos .....	16
3.1.1.	<i>Branch and Bound</i> .....	16
3.1.2.	<i>Branch and Cut</i> .....	17
3.2.	Heurísticas .....	18
3.2.1.	<i>Clarke and Wright</i> .....	18
3.2.2.	<i>Fisher and Jaikumar</i> .....	19
3.2.3.	<i>Matching Based</i> .....	20
3.2.4.	<i>Multi-Route Improvement</i> .....	20
3.2.5.	<i>Sweep</i> .....	20
3.2.6.	<i>Petal</i> .....	21
3.3.	Meta-heurísticas .....	22
3.3.1.	<i>Ant System</i> .....	22
3.3.2.	<i>Simulated Annealing</i> .....	23
3.3.3.	Algoritmos Genéticos .....	24
3.3.4.	<i>Tabu Search</i> .....	24
3.3.5.	<i>Neural Network</i> .....	25
3.4.	Aproximações multiobjectivos .....	26
4.	Sistema de Informação Geográfica de apoio à decisão .....	27
4.1.	Enquadramento .....	28
4.2.	Características .....	29
4.3.	Componentes .....	29
4.4.	A ferramenta SIG desenvolvida .....	31
4.4.1.	Pressupostos e funcionalidades .....	31

4.4.2.	O VRP .....	33
4.4.2.1.	Instâncias a resolver .....	33
4.4.2.2.	Soluções a aplicar .....	33
4.4.3.	Tecnologias.....	34
4.4.4.	Suporte de informação .....	35
4.4.5.	A aplicação .....	39
4.4.5.1.	Povoamento e gestão do sistema de informação .....	41
4.4.5.1.1.	Ecrãs de gestão.....	41
4.4.5.1.2.	Inserção de registos, a matriz de distâncias e os <i>clusters</i> .....	42
4.4.5.1.3.	Visualização, edição e eliminação de registos.....	45
4.4.5.1.4.	Listagem de registos.....	46
4.4.5.2.	Funcionalidades de apoio à decisão .....	48
4.4.5.2.1.	Percurso entre dois pontos .....	48
4.4.5.2.2.	Rota Manual.....	49
4.4.5.2.3.	Rotas Automáticas.....	51
4.4.5.2.3.1.	Algoritmo 1 .....	52
4.4.5.2.3.2.	Algoritmo 2 .....	54
4.4.5.2.3.3.	Algoritmo de Pesquisa Local – <i>Best Improvement</i> .....	56
4.4.5.2.3.4.	Exemplo de Execução 1 .....	58
4.4.5.2.3.5.	Exemplo de Execução 2.....	63
4.4.5.2.3.6.	Exemplo de Execução 3.....	65
4.4.5.2.3.7.	Exemplo de Execução 4.....	69
4.4.5.2.4.	Visualizar/Apagar Rotas.....	72
4.4.5.2.5.	Comparar resultados .....	73
5.	Conclusão.....	75
5.1.	Trabalho futuro.....	76
	Bibliografia.....	77
	ANEXOS.....	81
	ANEXO 1 – Pseudo-código do Algoritmo 1.....	82
	ANEXO 2 – Pseudo-código do Algoritmo 2.....	83
	ANEXO 3 – Pseudo-código do Algoritmo de Pesquisa Local .....	84

ANEXO 4 – *Challenge* InoCrowd – Redução de consumos de combustível ..... 85

## Lista de siglas/abreviaturas

---

**AG** Algoritmo Genético, 24

**BPP** *Bin Packing Problem* (Problema de Empacotamento), 15, 16

**CVRP** *Capacitated Vehicle Routing Problem* (Problema de Encaminhamento de Veículos de Capacidades fixas), 11, 17, 20, 21, 23, 33

**DER** Diagrama de Entidades e Relações, 35, 37

**MDVRP** *Multiple Depot Vehicle Routing Problem* (Problema de Encaminhamento de Veículos com Múltiplos Depósitos), 12, 23, 33

**MTSP** *Multiple Traveling Salesman Problems* (Problema do Caixeiro Viajante Múltiplo), 15

**PVRP** *Periodic Vehicle Routing Problem* (Problema de Encaminhamento de Veículos com Periodicidade), 13

**SDVRP** *Split Delivery Vehicle Routing Problem* (Problema de Encaminhamento de Veículos com Entregas Distribuídas), 12, 25

**SIG** Sistema de Informação Geográfica, iv, 1, 2, 3, 27, 28, 29, 30, 31, 32, 33, 34, 39, 75

**SVRP** *Stochastic Vehicle Routing Problem* (Problema de Encaminhamento de Veículos Estocástico), 13

**TDMC** Tomada de decisão multicritério, 28

**TS** *Tabu Search* (Pesquisa Tabu), 24, 25

**TSP** *Traveling Salesman Problem* (Problema do Caixeiro Viajante), 15, 16, 19, 20

**VRP** *Vehicle Routing Problem* (Problema de Encaminhamento de Veículos), sigla presente em grande parte das páginas e por isso as mesmas não são especificadas

**VRPB** *Vehicle Routing Problem with Backhauls* (Problema de Encaminhamento de Veículos com Lanços de Volta), 11, 12, 20



**VRPPD** *Vehicle Routing Problem with Pick-up and Delivery* (Problema de Encaminhamento de Veículos com Recolha e Entrega), 12

**VRPTW** *Vehicle Routing Problem with Time Windows* (Problema de Encaminhamento de Veículos com Janelas de Tempo), 7, 11, 16

## Índice de Figuras

---

Figura 1 – Representação típica de um problema VRP .....	6
Figura 2 - Exemplo de solução de problema de definição de rotas .....	6
Figura 3 - Conceito de poupanças (Lysgaard, 1997).....	19
Figura 4 – Diagrama de Entidades e Relações para sistema de informação.....	36
Figura 5 – Página inicial do SIG de apoio à decisão .....	39
Figura 6 – Ecrã de gestão da entidade <i>Encomendas</i> .....	42
Figura 7 – Conjunto de ecrãs exemplificativos da operação de inserção de registo .....	43
Figura 8 – Conjunto de ecrãs exemplificativos das operações de visualização, edição e eliminação de registo.....	45
Figura 9 – Exemplos de listagens .....	47
Figura 10 – Percorso entre dois pontos .....	48
Figura 11 – <i>Inputs</i> iniciais da funcionalidade <i>Rota Manual</i> .....	49
Figura 12 – Funcionalidade <i>Rota Manual</i> .....	50
Figura 13 – Ecrã de <i>inputs</i> para rotas automáticas .....	51
Figura 14 – <i>Algoritmo 1</i> : Todos os clientes com encomendas .....	59
Figura 15 – <i>Algoritmo 2</i> : Todos os clientes com encomendas .....	60
Figura 16 – Pesquisa local aplicada ao <i>Algoritmo 1</i> : Todos os clientes com encomendas .	62
Figura 17 – Listagem de encomendas que servirão de <i>input</i> para os algoritmos no segundo exemplo de execução .....	63
Figura 18 – <i>Algoritmo 1</i> aplicado ao segundo <i>input</i> de testes.....	64
Figura 19 – <i>Algoritmo 2</i> aplicado ao segundo <i>input</i> de testes.....	64
Figura 20 – Listagem de encomendas que servirão de <i>input</i> para os algoritmos no terceiro exemplo de execução .....	66
Figura 21 – <i>Algoritmo 1</i> aplicado ao terceiro <i>input</i> de testes .....	66
Figura 22 – <i>Algoritmo 2</i> aplicado ao terceiro <i>input</i> de testes .....	67

Figura 23 – <i>Best Improvement</i> aplicada ao resultado do terceiro <i>input</i> de testes do <i>Algoritmo 1</i> .....	68
Figura 24 – <i>Best Improvement</i> aplicada ao resultado do terceiro <i>input</i> de testes do <i>Algoritmo 2</i> .....	69
Figura 25 – <i>Algoritmo 1</i> aplicado ao quarto <i>input</i> de testes.....	70
Figura 26 – <i>Algoritmo 2</i> aplicado ao quarto <i>input</i> de testes.....	71
Figura 27 – Conjunto de ecrãs descritivos da funcionalidade de <i>Visualizar/Apagar Rota</i> ...	72
Figura 28 – Ecrã representativo da funcionalidade.....	74
<i>Comparação de Algoritmos</i> .....	74

# 1. Introdução

---

## 1.1. Motivação

Em 2008 a Comissão Europeia estimava que o sector da logística representava um montante de cerca de 5400 mil milhões de euros a nível mundial, ou seja, cerca de 13,8 % do PIB mundial. Em média, os custos de logística representavam nessa altura entre 10 e 15 % do custo final dos produtos acabados. Porém, foi precisamente em 2008 que se deu um forte choque petrolífero e, desde então, a escalada dos preços dos combustíveis catapultou os custos associados ao transporte de bens para níveis ainda superiores, que levam a que o problema de definição de rotas assuma grande importância. Neste âmbito, ganham força as soluções para os problemas de encaminhamento de veículos ou VRP, soluções essas que se propõem encontrar rotas otimizadas tendo em conta, em termos básicos, os veículos disponíveis, a localização do(s) depósito(s) e dos clientes.

Paralelamente ao aumento abrupto do preço dos combustíveis surgem os avanços tecnológicos, nomeadamente de sistemas e tecnologias informáticas. Ferramentas como os SIG, conhecidas há vários anos (mais de duas décadas) por permitirem o armazenamento, gestão e tratamento de dados textuais e georreferenciados, podem agora ser municiadas e potenciadas com mais-valias *Web*, como por exemplo as fornecidas pelo conhecido *Google Maps*, que permitem a obtenção e atualização de informações georreferenciadas, como distâncias e estimativas de tempos de percurso entre pontos em tempo real, assim como o desenho de mapas e rotas interativas em aplicações próprias.

## 1.2. Objetivo

O principal objetivo desta dissertação é, tendo por base as motivações acima enunciadas, a construção de um SIG baseado na *Web* que apoie a tomada de decisão de um decisor humano,

como por exemplo um gestor de frota, no âmbito de um VRP genérico, ou seja, um VRP que se adequa a um número razoável de casos reais ou que possa facilmente ser adaptado a tal.

A ferramenta informática a desenvolver deverá permitir albergar informação textual e georreferenciada sobre os constituintes da problemática, como por exemplo depósitos, clientes, veículos e encomendas. Munido destas informações o SIG deverá fornecer ao decisor ferramentas que suportem as suas opções, como por exemplo, a visualização de informações sobre rotas que o mesmo tenha definido ou a apresentação de sugestões de rotas a realizar. Para o cálculo destas rotas a sugerir, o sistema deve estar capacitado com algoritmos heurísticos de resolução de VRP.

### **1.3. Visão Global**

Como se consegue antever através do objetivo apresentado, esta dissertação baseia-se num trabalho de investigação-ação cuja temática alvo gravita em torno do problema de encaminhamento de veículos. Na primeira fase, a de investigação, efetuar-se-á um estudo aprofundado sobre os VRP, nomeadamente sobre os seus elementos constituintes, objetivos, instâncias e algoritmos de resolução.

Numa fase de transição entre a investigação e a ação avançar-se-á para a escolha das tecnologias de desenvolvimento de *software* a utilizar, para a decisão e estruturação do Sistema de Informação que suportará a aplicação e para uma análise do funcionamento e das funcionalidades oferecidas pela interface *Google Maps*. Ainda numa fase anterior à ação decidir-se-ão quais os tipos de VRP que a aplicação se proporá resolver e quais as formas de resolução a aplicar.

Após a consolidação destes pontos avançar-se-á para a fase da ação propriamente dita, com o desenvolvimento do SIG apoiado na *Web* de auxílio à decisão para VRP.

### **1.4. Estrutura da dissertação**

A dissertação divide-se em cinco capítulos: *Introdução*, *Vehicle Routing Problem*, *Algoritmos para resolução de VRP*, *Sistema de Informação Geográfica de apoio à decisão* e *Conclusão*. Neste primeiro capítulo descrevem-se em linhas gerais as motivações para a

realização de uma dissertação sobre os VRP direcionada para a construção de uma ferramenta de apoio à decisão nesse âmbito. Apresenta-se também o objetivo que se persegue e uma visão global sobre o trabalho a realizar. Conclui-se o capítulo com a exposição da estrutura deste documento.

O segundo capítulo iniciar-se-á com realização de um enquadramento da problemática, apresentando de forma detalhada os motivos para a realização de uma investigação sobre o assunto. De seguida avançar-se-á para uma pesquisa bibliográfica sobre o estado da arte no âmbito dos VRP, caracterizando-se os seus constituintes, os objetivos genéricos que existem para a sua solução e quais as principais instâncias de VRP bibliograficamente reportadas.

No terceiro capítulo, realizar-se-á um estudo sobre os principais algoritmos exatos, heurísticas, meta-heurísticas e aproximações multiobjectivos existentes, associando cada um destes algoritmos às instâncias VRP a que se aplicam e que resolvem.

O quarto capítulo marcará a transição entre a investigação e a ação e apresentará a implementação do trabalho de ação propriamente dito. Este iniciar-se-á com uma investigação sumária sobre os SIG, suas características e componentes e com a apresentação das mais-valias da sua utilização para apoio à decisão em VRP. De seguida avançar-se-á para o começo da instanciação da problemática a resolver, designadamente assumindo os pressupostos e as funcionalidades consideradas necessárias, os tipos de VRP a trabalhar e os algoritmos de resolução a aplicar. Antes de se iniciar a apresentação dos elementos da aplicação propriamente ditos, fundamentar-se-ão as tecnologias de desenvolvimento de *software* apropriadas para o caso e apresentar-se-á também o estudo e construção do Sistema de Informação que formará o suporte do conhecimento que a aplicação manipulará. Por fim avançar-se-á para a apresentação das funcionalidades da aplicação, começando pelas de povoamento do sistema de informação ou, por outras palavras, de caracterização do VRP, e avançando para as de apoio à decisão. Esta exposição demonstrativa da aplicação será documentada com imagens e esclarecimentos sobre cada funcionalidade desenvolvida.

Concluir-se-á este relatório realizando uma reflexão sobre as potencialidades da ferramenta construída, avaliando o cumprimento dos objetivos e lançando propostas de mais-valias passíveis de serem acrescentadas futuramente.



## ***2. Vehicle Routing Problem***

---

Neste capítulo começar-se-á por efetuar um enquadramento, informando o leitor de quais as motivações e relevância em termos académicos da problemática, que levam a que seja pertinente a realização de uma investigação aprofundada sobre o assunto.

Após este elemento impulsionador do interesse, avançar-se-á para a caracterização dos elementos básicos dos VRP e dos seus objetivos gerais.

### **2.1. Enquadramento**

Atualmente, com a concorrência que marca todas as áreas empresariais, o sucesso da comercialização passa pela atenção despendida a todos os aspetos que permitam reduzir custos, estando estes inversamente relacionados com a competitividade dos produtos no mercado. Relativamente ao transporte de bens, que constituirá a problemática alvo desta dissertação, em 2008 a Comissão Europeia estimava que o sector da logística representava um montante de cerca de 5400 mil milhões de euros a nível mundial, ou seja, cerca de 13,8 % do PIB mundial. Em média, os custos da logística representavam entre 10 e 15 % do custo final dos produtos acabados. Porém, como é de conhecimento geral, desde o choque petrolífero de 2008 que os preços dos combustíveis tem vindo a aumentar abruptamente, catapultando os custos associados à logística para valores ainda superiores. Neste sentido, empresas que possuam uma frota de veículos têm todo o interesse em reduzir os custos que lhe estão associados, sendo que o problema de definição de rotas assume grande importância, uma vez que os valores monetários que lhe estão agregados, nomeadamente de combustível, de manutenção da frota e outros que se analisarão posteriormente são avultados e mantêm-se num crescimento para o qual não se avista um fim.

Neste âmbito, surgem as soluções para os problemas de encaminhamento de veículos ou VRP, soluções essas que se propõem encontrar rotas otimizadas tendo em conta, em termos básicos, os veículos disponíveis, a localização do(s) depósito(s) e dos clientes. Ou seja, à partida



tem-se uma representação semelhante à apresentada na Figura 1, e o objetivo é obter uma solução para o VRP, em termos das rotas a realizar pelos veículos que saem do depósito da empresa, realizam entregas ou recolhas de produtos e retornam, que minimize os gastos inerentes a esse processo.

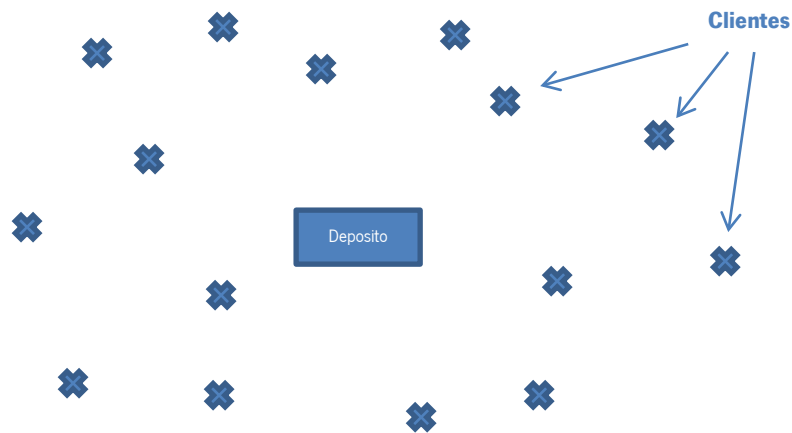


Figura 1 – Representação típica de um problema VRP

Pretende-se então, partindo de um problema com uma estrutura genérica semelhante à da Figura 1 chegar à melhor solução possível das trajetórias a realizar, conforme exemplo apresentado na Figura 2.

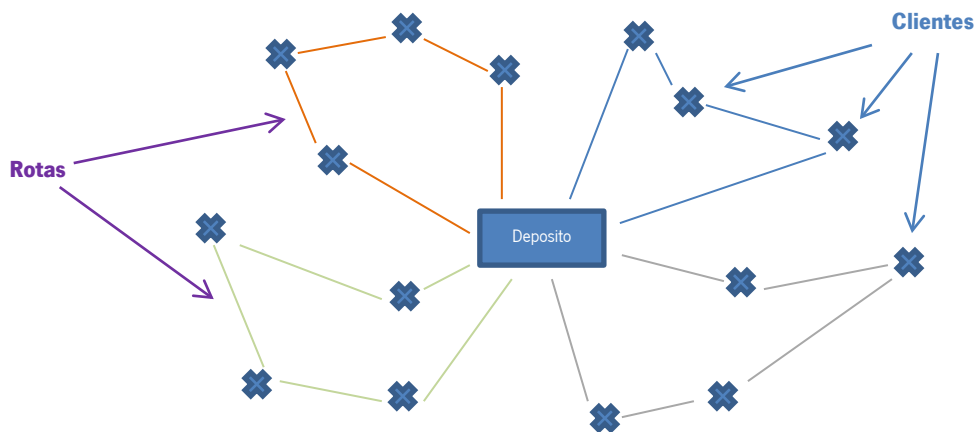


Figura 2 - Exemplo de solução de problema de definição de rotas

Este não é um problema recente, tendo sido introduzido por Dantzig e Ramser em 1959, com a utilização de algoritmos de programação matemática para a resolução de um caso real de

distribuição de gasolina por estações de serviço. Alguns anos mais tarde (1964), Clarke e Wright implementaram uma heurística *greedy*, ou seja, que em cada iteração escolhe a melhor opção, tendo esta solução melhorado a de Dantzig-Ramser. Depois destes percursores surgiram variados modelos e algoritmos para soluções ótimas e aproximadas para VRP, existindo diversas aplicações para casos reais no mercado.

Em 2002, Toth & Vigo referiam que, “o interesse nos VRP é motivado pela sua relevância prática e a sua dificuldade considerável de implementação: as maiores instâncias de VRP que podem ser consistentemente resolvidas pelos algoritmos exatos mais eficientes existentes no momento contêm 50 clientes, sendo que problemas maiores podem ser resolvidos otimamente apenas em casos particulares” (p. xvii). Já em 2012, Lin, Chou, Lee, & Lee dizem que “o VRPTW [um tipo de VRP que se analisará posteriormente] é NP-completo e instâncias com 100 clientes ou mais são muito difíceis de resolver otimamente” (p. 11). Tendo em conta estes testemunhos e o enunciado relativamente simples dos VRP, o que os torna difíceis de solucionar? Ralphs, Hartman, & Galati (2001) afirmam que uma das razões é o facto de este tipo de problemas ser a interseção de dois problemas difíceis: *Traveling Salesman Problem* (Problema do Caixeiro Viajante - encaminhamento ou roteamento); e o *Bin Packing Problem* (empacotamento).

No mundo real a aplicabilidade deste tipo de problemas vai muito além da entrega e recolha de produtos entre empresas e seus clientes. Exemplos como a recolha de resíduos, limpeza de ruas, rotas de autocarros escolares, transportes aéreos, transporte de deficientes, rotas de vendedores e de veículos de manutenção, são casos reais da aplicabilidade de soluções de VRP.

## 2.2. Conceitos básicos

Conforme foi referido em termos simplificados anteriormente e esmiuçando agora um pouco mais, é possível afirmar que a solução para um VRP passa por, dado um conjunto de **veículos** que à partida estão num ou mais **depósitos**, a serem conduzidos por um conjunto de **condutores**, um conjunto de locais a visitar, tipicamente **clientes**, e uma **rede rodoviária** existente, determinar quais as **rotas** a realizar de forma que cada veículo retorne ao depósito de onde saiu, as necessidades dos clientes fiquem satisfeitas, as restrições operacionais sejam cumpridas e os custos globais de transporte sejam minimizados.

Tendo estas premissas e associações em conta, avançar-se-á para a descrição das características de cada um destes elementos constituintes dos VRP, base a partir da qual se formularão posteriormente restrições e casos particulares do problema em análise.

### **2.2.1. Rede rodoviária**

Como é natural, as ligações entre os clientes e entre estes e os depósitos são realizadas através de uma rede rodoviária que pode ser descrita como sendo um grafo onde os clientes e os depósitos são os vértices e as estradas são os arcos, direcionais ou não, dependendo das estradas em questão, sendo que a cada arco está associado um custo que representa o tempo de deslocação, distância, entre outros.

### **2.2.2. Clientes**

Os clientes são normalmente caracterizados, conforme referido no ponto anterior, como sendo vértices do grafo (rede rodoviária) e têm a si associada uma quantidade de bens, que podem ser diferentes, a serem entregues ou recolhidos, havendo também a possibilidade de se levarem uns artigos e trazerem-se outros. Podem também ter de ser servidos num dado período de tempo devido a horários de abertura ou limitações de tráfego, tendo ainda a si associados tempos de descarga/carga. Pode ainda dar-se o caso de haver clientes que têm que ser servidos por veículos específicos, dadas as acessibilidades rodoviárias ou de descarga/carga.

Há ainda casos em que clientes não conseguem ser totalmente servidos, havendo a necessidade de repartir a encomenda por mais do que uma viagem. Este tipo de situações pode dar lugar a custos de penalizações, a serem associadas aos clientes.

### **2.2.3. Depósitos**

Cada veículo sai e retorna ao mesmo depósito, sendo este caracterizado pelo número de veículos que alberga e pela quantidade de produtos que consegue despachar por período de tempo. Os depósitos são representados como sendo vértices do grafo (tal como os clientes).

Existem casos de VRP com múltiplos depósitos porém, como referem Toth & Vigo, “em algumas aplicações no mundo real, os clientes são à priori divididos pelos depósitos, e os veículos têm que retornar ao depósito de onde saíram, no final de cada rota. Nestes casos, o VRP pode ser decomposto em vários problemas independentes, sendo que cada um está associado a um depósito diferente” (2002, pp. 2,3). Como se verá posteriormente, aos espaços geográficos correspondentes à referida divisão chamar-se-ão *clusters*.

#### **2.2.4. Veículos**

Os veículos que permitem o transporte dos bens são caracterizados por terem um depósito a que estão associados e a onde devem voltar depois de cada serviço (rota), pela sua capacidade (total ou parcial para cada tipo de produto que podem levar), se têm ou não componentes que facilitem a carga/descarga, se existem estradas (arcos) em que não possam passar e quais e os custos associados à sua utilização (por distância, por tempo, por rota, etc.).

#### **2.2.5. Condutores**

Os condutores estão sujeitos a variadas restrições que se prendem com fatores pessoais, contratuais e sindicais como por exemplo que veículos podem conduzir, o número de horas de trabalho diárias, horário de trabalho e tempo máximo de condução seguido.

#### **2.2.6. Rotas**

As características das rotas estão intimamente ligadas às de todos os elementos até aqui descritos. Estas são compostas por um depósito e um conjunto de clientes, vértices do grafo, e pelas ruas (arcos) que unem os vértices. Existem ainda diversas restrições associadas às rotas, como por exemplo o facto de o total de encomendas dos clientes que as compõem ter que ser igual ou inferior à capacidade dos veículos que as servem, tendo em consideração se as rotas são apenas de distribuição ou recolha, ou de distribuição e recolha. Também poderá haver precedências que impõem a ordem pela qual os clientes devem ser servidos, principalmente nos casos de distribuição e recolha, uma vez que o veículo poderá ter que distribuir carga antes de

fazer alguma coleta, e o horário em que os clientes podem receber as encomendas terá igualmente de ser tido em conta.

Para o cálculo dos custos globais mínimos associados a cada rota é necessário o conhecimento dos custos de viagem entre cada par de clientes e entre cada cliente e o depósito, tendo em conta as restrições operacionais de cada caso.

### **2.3. Objetivos**

Conforme anteriormente referido, o objetivo primário da solução dos VRP é a redução de custos. Porém existem outros que, podendo ser combinados com este ou não, devem ser considerados. Exemplos de outros objetivos são a minimização do número de veículos necessários para servir os clientes, o equilíbrio das rotas em termos de tempo de viagem ou de carga dos veículos e a minimização das penalizações devidas a atrasos na entrega.

Os objetivos dos VRP variam consoante a natureza do problema e, assim sendo, serão clarificados de seguida em paralelo com a exposição das instâncias de VRP que se consideram de maior relevo.

### **2.4. Instâncias**

Dada a grande abrangência de problemas deste tipo, têm sido estudadas diversas variações, ou casos particulares de VRP. Estas variações estão associadas com os vários tipos de restrições que são vulgares na transposição para a prática deste tipo de problemas. Utilizando como fio condutor a fonte de informação electrónica VRP Web (Auren & NEO, 2007), enunciar-se-ão de seguida as instâncias de VRP que se consideram de maior relevo, bem como as soluções para cada caso específico que podem ser agrupadas entre si, compondo resoluções para questões mais complexas.

### **2.4.1. *Capacitated VRP (CVRP)***

O CVRP é “o membro mais simples e mais estudado” (Toth & Vigo, 2002, p. 5) dos VRP. A principal restrição implícita neste caso é de que cada veículo da frota tem uma capacidade limitada e uniforme.

O CVRP pressupõe ainda uma frota de tamanho fixo, encomendas conhecidas de uma só mercadoria e a existência de um único depósito sendo que o objetivo deste tipo de VRP é “minimizar o custo total (p.e., numa função ponderada do número de rotas e do seu comprimento ou tempo de percurso) de serviço a todos os clientes” (Toth & Vigo, 2002, p. 5).

### **2.4.2. *VRP with Time Windows (VRPTW)***

É o caso de VRP onde todos os clientes têm que ser atendidos tendo em conta janelas temporais. Segundo referem Bräysy & Gendreau, investigadores das universidades de Jyväskylä (Finlândia) e de Montréal (Canadá) respetivamente, num VRPTW “as rotas devem ser definidas de tal maneira que cada ponto é visitado apenas uma vez, por apenas um veículo, dentro de um dado intervalo de tempo, sendo que todas as rotas se iniciam e concluem no depósito e o total de encomendas de cada rota não pode exceder a capacidade do veículo que a serve” (2005, p. 104).

Toth & Vigo (2002) definem o VRPTW como sendo uma extensão do CVRP com o acréscimo do intervalo de tempo associado a cada cliente.

### **2.4.3. *VRP with Backhauls (VRPB)***

O VRPB é o caso de VRP onde os clientes além de receberem produtos, também têm carga a ser devolvida ao depósito (Jacobs-Blecha & Goetschalckx, 1998).

Sobre os VRPB, Toth & Vigo afirmam que “as restrições associadas às operações de carga e descarga, e a dificuldade de rearranjo da carga do veículo durante a rota, significam que todas as entregas devem ser realizadas antes das recolhas” (2002, p. 9).

#### **2.4.4. Multiple Depot VRP (MDVRP)**

Segundo Carlsson, Ge, Subramaniam, Wu, & Ye, da Universidade de Stanford, o *Multiple Depot VRP* é a “generalização do *Single Depot VRP* onde os múltiplos veículos existentes partem de diferentes depósitos e voltam para os seus depósitos originais no final das rotas que lhe foram atribuídas” (2007, p. 1). Nesta definição resta clarificar que o *Single Depot VRP* mencionado corresponde ao VRP onde existe apenas um depósito de onde todos os veículos saem e para onde retornam.

Os últimos autores citados afirmam no mesmo trabalho que “o objetivo tradicional do MDVRP é de minimizar a soma de todos os comprimentos” (pp. 1,2), onde por comprimentos entendemos ser as distâncias percorridas pelos veículos.

#### **2.4.5. VRP with Pick-up and Delivery (VRPPD)**

Esta variante de VRP, que permite a recolha e entrega de material, surgiu para dar resposta a situações de casos reais, como por exemplo, o da recolha de resíduos para a reciclagem, além da distribuição dos produtos. A primeira abordagem mostra-nos muitas semelhanças entre o VRP *with Pick-up and Delivery* e o VRP *with Backhauls*, problema onde também existia distribuição e recolha de bens. Porém, Righini (2000) clarifica os conceitos informando que, tal como já tinha sido referido anteriormente, no VRPB os veículos devem realizar todas as entregas antes de iniciar as recolhas, sendo que no VRPPD não existe esta restrição, ou seja, podem existir recolhas antes de estarem todas as encomendas distribuídas o que, pelo raciocínio induzido por Toth & Vigo (2002) na análise do VRPB, produzirá um problema de resolução mais difícil no caso do VRPPD.

#### **2.4.6. Split Delivery VRP (SDVRP)**

No *Split Delivery VRP* (também referido na literatura por *Split Delivery Problem*) os clientes podem ser servidos por mais que um veículo, ou seja, é a formulação correta quando algumas encomendas de clientes são maiores do que a capacidade dos veículos. Archetti, Mansini, & Speranza, da Universidade de Brescia, apontam que o objetivo do SDVRP é minimizar o custo da distância total percorrida pelos veículos para servir todos os clientes (2003).

#### **2.4.7. Stochastic VRP (SVRP)**

“Em vários contextos práticos, um ou vários componentes do problema são aleatórios. Esta situação faz surgir os *Stochastic VRP*” (Laporte & Louveaux, 1997, p. 1). Dada a definição é fácil enunciar os componentes que poderão ser aleatórios. Poderemos falar do número de clientes, do tamanho das encomendas, do tempo de serviço ou do tempo de viagem.

#### **2.4.8. Periodic VRP (PVRP)**

Tipicamente, esta é a variação de VRP em que as entregas podem ser periódicas. Coene, Arnout, & Spieksma, no seu caso de estudo sobre uma companhia de recolha de lixo belga, afirmam que no PVRP “existe um horizonte, digamos de  $T$  dias, e existe uma frequência que dita, para cada cliente, quantas vezes este tem que ser visitado nesses  $T$  dias” (2008, p. 1), sendo que, segundo o mesmo trabalho, o objetivo da solução deste tipo de problemas é obter um conjunto de rotas que satisfaçam tanto as encomendas como a frequência de visitas, minimizando a soma dos custos de todas as rotas para o período de tempo planeado.

Estes autores referem ainda que a solução deste tipo de problemas assenta “em empresas de reparações periódicas ou empresas que efetuem recolha/distribuição de bens periodicamente”.





### 3. Algoritmos para resolução de VRP

---

Neste capítulo abordar-se-ão as técnicas mais utilizadas para a resolução dos VRP. Desde a apresentação do primeiro VRP por Dantzig e Ramzer, muitos algoritmos foram propostos para resolver este tipo de problemas e os que de si derivam. Entre estes é possível diferenciar os algoritmos exatos, que procuram apenas soluções ótimas, dos algoritmos heurísticos, que procuram soluções aproximadas, trocando exatidão por benefícios que se explicarão de seguida.

Dado este tipo de problemas ser NP-difícil, “o esforço computacional requerido para a sua resolução cresce exponencialmente com a dimensão do problema” (Auren & NEO, 2007), sendo que a dimensão destes na literatura é caracterizada e medida principalmente através do número de clientes. Para a maioria deste tipo de problemas são toleráveis e até muitas vezes desejáveis, soluções aproximadas que sejam simultaneamente suficientemente precisas e encontradas num período de tempo considerado aceitável, sendo para estes casos invocados os algoritmos ou métodos heurísticos ou meta-heurísticos.

Conforme anteriormente referido, a dificuldade dos VRP advém do facto de estes serem a composição de dois problemas difíceis: o *Traveling Salesman Problem* e o *Bin Packing Problem*. Passamos a explicar a relação entre estes problemas de optimização, baseando-nos em informações obtidas de Auren & Network and Emerging Optimization (2007):

- *Traveling Salesman Problem* (TSP): se no VRP a capacidade dos veículos fosse infinita estes corresponderiam a instâncias de *Multiple Traveling Salesman Problems* (MTSP), problema que por sua vez poderia ser transformado em TSP através da inserção no grafo de cópias do depósito (tantas quantas o número de rotas menos 1).
- *Bin Packing Problem* (BPP): como a capacidade dos veículos é limitada, a questão de saber se existe ou não uma solução possível para uma instância de VRP é uma instância de BPP (associada às capacidades de carga dos veículos), não sendo considerados neste caso os custos associados às arestas do grafo (são igualados a zero). Assim, todas as soluções possíveis têm o mesmo custo.

Desta forma chega-se à razão da afirmação de que os VRP são a composição de TSP com BPP: a solução do problema é composta pela definição das rotas (TSP), tendo em conta restrições de acondicionamento (BPP). Devido aos VRP serem a composição destes dois problemas, a solução ótima de algumas das suas instâncias pode ser extremamente difícil de encontrar.

Avança-se de seguida para a apresentação das principais técnicas de resolução para VRP.

### **3.1. Algoritmos exatos**

Os algoritmos exatos, por procurarem apenas soluções ótimas, têm normalmente um custo computacional de processamento elevado e, por isso, são geralmente utilizados apenas para instâncias de menor dimensão. Invocando citações anteriores, dada a sua relevância para o assunto em questão, é de lembrar que, em 2002, Toth & Vigo referiam que “as maiores instâncias de VRP que podem ser consistentemente resolvidas pelos algoritmos exatos mais eficientes existentes no momento contêm 50 clientes, sendo que problemas maiores podem ser resolvidos otimamente apenas em casos particulares” (p. xvii). Mais recentemente, em 2012, Lin *et al.* afirmam que “o VRPTW é NP-completo e instâncias com 100 clientes ou mais são muito difíceis de resolver optimamente” (p. 11). Tendo os aspetos gerais relativos aos algoritmos exatos especificados, prosseguir-se-á de seguida para as principais instâncias destes algoritmos.

#### **3.1.1. *Branch and Bound***

O *Branch and Bound* (Partição e Avaliação Sucessivas) é conhecido por ser um método de pesquisa em árvore com enumeração e limitação, e consiste em dividir o problema principal sucessivamente em conjuntos de problemas menores, calculando limites para o valor da função objetivo em cada um desses subconjuntos. Esses limites são utilizados para descartar alguns subconjuntos de problemas menores de outras iterações.

Segundo Carravilla & Oliveira (2010), este tipo de problemas enquadram-se nas técnicas exatas para resolução de problemas de optimização combinatoria, caracterizando-se como sendo de enumeração implícita pois, não se gerando todas as soluções admissíveis, elas são no entanto consideradas e implicitamente avaliadas.

O algoritmo de *Branch and Bound* tem sido amplamente estudado ao longo dos anos. No âmbito dos VRP este tem sido aplicado por exemplo em VRP simples com regras de portagens por peso (Zhang, Qin, Lim, & Guo, 2010), em CVRP para grafos direcionais (Fischetti, Toth, & Vigo, 1994) e em VRP com restrições de distâncias assimétricas (Laporte, Nobert, & Taillefer, 1987).

### **3.1.2. *Branch and Cut***

Este algoritmo tem sido utilizado na resolução de problemas interessantes como a definição de rotas de autocarros escolares, e de VRP com instalações satélite, que permitem que os condutores reabasteçam e continuem a efetuar distribuições sem voltar ao depósito (Bard, Huang, Dror, & Jaillet, 1998).

### 3.2. Heurísticas

As heurísticas clássicas que se descreverão neste ponto vieram dar resposta à impraticabilidade dos algoritmos exatos para problemas reais de grande dimensão, tendo surgido entre 1960 e 1990. Dado que estes métodos geralmente produzem bons resultados em curto espaço de tempo computacional e podem ser facilmente ajustados às restrições encontradas na realidade, ainda se mantêm amplamente utilizados em pacotes de apoio à decisão comerciais (Laporte, Gendreau, Potvin, & Semet, 2000).

“As heurísticas de encaminhamento de veículos, como a maioria das heurísticas, são normalmente medidas através de dois critérios: precisão e velocidade. Na nossa opinião simplicidade e flexibilidade são também atributos essenciais de boas heurísticas” (Cordeau, Gendreau, Laporte, Potvin, & Semet, 2002, p. 513). No seguimento desta citação, que aponta as características a ter em conta na análise de heurísticas, avançar-se-á para a apresentação das principais.

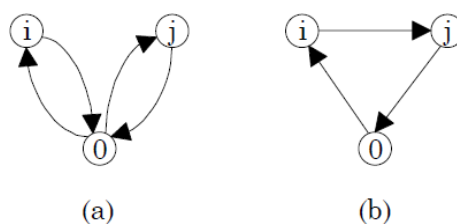
Os métodos heurísticos podem ser divididos, segundo Laporte & Semet (2002), em três categorias. As **heurísticas construtivas** vão construindo gradualmente uma solução possível, mantendo atenção ao custo da solução, mas não contêm uma fase de melhoria em si. Nas **heurísticas de duas fases**, o problema é decomposto nos seus dois componentes naturais: agrupamento dos vértices em rotas possíveis e construção da rota ótima atual, com a possível repetição destes processos. Finalmente, os **métodos de melhoria** tentam aperfeiçoar as soluções possíveis através de uma sequência de trocas de vértices ou ligações entre ou dentro das rotas.

#### 3.2.1. *Clarke and Wright*

Este clássico algoritmo construtivo que tomou o nome dos seus autores foi publicado em 1964 e baseia-se no conhecido conceito de poupanças ou economias (*savings*).

Como método heurístico que é, o *Clarke and Wright* não garante uma solução ótima, porém oferece frequentemente uma boa solução, ou seja, uma solução com um pequeno desvio da ótima (Lysgaard, 1997).

O conceito de poupanças anteriormente referido expressa o custo das poupanças obtido pela junção de duas rotas em uma, conforme apresentado na figura abaixo, sendo que 0 representa o depósito e  $i$  e  $j$  representam clientes.



**Figura 3 - Conceito de poupanças (Lysgaard, 1997)**

O custo associado ao conjunto de rotas (a) seria:

$$Custo (a) = c_{0i} + c_{i0} + c_{0j} + c_{j0}$$

O custo associado ao conjunto de rotas (b) seria:

$$Custo (b) = c_{0i} + c_{ij} + c_{j0}$$

Pelo que as poupanças combinadas por fazer apenas a rota (b) seriam:

$$Poupança (a) = Custo (a) - Custo (b)$$

Sobre este método, Teixeira & Cunha afirmam que “muitos dos pacotes comerciais tem como base a heurística de economias, proposta por Clarke e Wright (1964). Além da rapidez em termos de tempo de processamento, a heurística de economias apresenta uma qualidade essencial no caso de um aplicativo comercial genérico: robustez, ou seja, capacidade de resolver satisfatoriamente problemas com diferentes restrições, sem degradação sensível na qualidade das soluções e nos tempos de processamento” (2002, p. 4). No seguimento desta informação, Cordeau *et al.* afirmam que este algoritmo “é um dos mais conhecidos e mantém-se largamente utilizado nos dias de hoje” (2002, p. 514). Estes autores referem ainda que o *Clarke and Wright* tem uma simplicidade e velocidade muito alta, apontando a falta de flexibilidade como sendo a sua maior falha.

### **3.2.2. Fisher and Jaikumar**

Nesta heurística datada de 1981, que também tomou o nome dos seus autores, o número de rotas é definido à partida, sendo efetuadas partições (cones) do plano de acordo com o peso dos clientes. Depois de realizadas as partições e formados os grupos de clientes por veículo realizando a resolução de um problema de atribuição generalizado, são resolvidos os TSPs

respetivos (Laporte & Semet, 2002). Na literatura este algoritmo também é conhecido por pertencer ao tipo *Cluster-first, route-second algorithms*.

Os autores anteriores apresentaram esta heurística como sendo uma das que resolve o CVRP. Além disso, o algoritmo de *Fisher and Jaikumar* foi também aplicado por Tan & Beasley (1984) num problema de definição de rotas em que nem todos os clientes recebem encomendas diariamente.

Cordeau *et al.*, no seu guia sobre heurísticas de VRP publicado em 2002, não consideram este algoritmo interessante ao nível da simplicidade, velocidade, precisão nem flexibilidade.

### **3.2.3. Matching Based**

A heurística construtiva denominada de *Matching Based* é apresentada por Desaulniers, Desrosiers, Erdmann, Solomon, & Soumis (2002) como solucionando o *VRP with Pickup and Delivery*. Os mesmos autores referem existir duas fases separadas na implementação deste método: uma primeira em que os clientes recebem as encomendas; e uma segunda em que são feitas as recolhas e estas são devolvidas ao depósito.

### **3.2.4. Multi-Route Improvement**

Esta técnica, descrita por Laporte & Semet (2002) como pertencendo ao grupo de heurísticas de melhorias que resolvem o CVRP, consiste em transferir ou trocar clientes entre rotas, de uma forma cíclica, procurando melhores resultados globais.

O conjunto de técnicas de *Improvement/Exchange*, onde se encontra o *Multi-Route Improvement* é descrito por Dubey como “podendo facilmente ser aplicado a soluções para VRPB, tendo em conta a relação de precedência das entregas antes das recolhas sempre que uma troca é considerada” (2010, p. 72).

### **3.2.5. Sweep**

Neste algoritmo são formados inicialmente *clusters* (agrupamentos) possíveis, traçando um raio a partir do depósito. É então obtida uma rota para cada *cluster*, através da resolução de um

TSP. É ainda possível efetuar uma otimização a este algoritmo, que consiste na permuta de vértices entre *clusters* adjacentes, de forma a verificar se as rotas globais ficam melhores (Laporte & Semet, 2002).

Este método é utilizado, por exemplo para a resolução do problema do tamanho de frota e de definição de rotas para veículos diferentes, abordado por Renaud & Boctor em 2002.

Segundo Cordeau *et al.* (2002), este algoritmo tem uma alta simplicidade, mas não parece ser superior ao algoritmo de *Clarke and Wright* em termos de precisão e velocidade, sendo também inflexível.

### **3.2.6. *Petal***

Os algoritmos *Petal*, pertencem ao grupo de métodos de duas fases e são das heurísticas clássicas de resolução do CVRP (Laporte & Semet, 2002). Os mesmos autores, na mesma obra, descrevem-nos como sendo uma “extensão natural do algoritmo *Sweep*” (p. 120), produzindo uma larga família de sobreposição de *clusters* (e rotas de veículos associadas) e selecionando de entre estes um conjunto possível de rotas, que denominam de *petals* (pétalas).



### **3.3. Meta-heurísticas**

As meta-heurísticas são uma classe de métodos que têm vindo a ser desenvolvidos desde o início da década de 80, tendo sido concebidas para resolver problemas onde as heurísticas não são eficazes nem eficientes (Osman & Laporte, 1996).

Os mesmos autores, na mesma obra de 1996, adiantam que uma meta-heurística é formalmente definida como um processo de geração iterativa, que guia a heurística subordinada através da combinação de diferentes conceitos inteligentes, pela exploração do espaço de pesquisa, sendo que estratégias de aprendizagem são utilizadas para estruturar a informação de modo a encontrar eficientemente soluções quase-óptimas.

Expor-se-ão de seguida as características gerais e as aplicações de várias meta-heurísticas.

#### **3.3.1. *Ant System***

A primeira meta-heurística que se apresenta vai buscar o seu nome à metáfora gerada pela associação entre a problemática em estudo e a forma como as formigas procuram comida e voltam para o seu ninho. Inicialmente as formigas exploram a área circundante ao seu refúgio de uma forma aleatória. Assim que uma formiga descobre uma fonte de comida, avalia o interesse da mesma em termos de qualidade e quantidade, e traz alguma da comida para o ninho. Durante o trajeto de retorno a formiga deixa no chão um rastro químico (feromona) em quantidade proporcional à qualidade da fonte de comida. O rastro servirá de guia para as outras formigas rumo à referida fonte e, estas alargarão o mesmo. Como as fontes que estão mais próximas do ninho são mais visitadas, os rastros destas é mais forte. O resultado final deste processo é a otimização do trabalho das formigas (Taillard, Gambardella, Gendreau, & Potvin, 2001).

Os mesmos autores, na mesma obra, criam as seguintes analogias entre a algoritmia de resolução de otimização combinatória e o comportamento das formigas:

- A área de procura das formigas e o conjunto de soluções possíveis do problema combinatório;
- A quantidade de comida associada à fonte com a função objetivo;
- O rasto e a memória adaptativa.

Um algoritmo *Ant System standard* tem a seguinte estrutura:

1. Inicializar o rastro;
2. Enquanto o critério de paragem não é verificado fazer:
  - a. Para cada formiga, construir uma nova solução utilizando o rastro atual e uma avaliação da solução parcial a ser construída;
  - b. Atualizar o rastro.

O componente mais importante de um *Ant System* é a gestão dos rastros. Nas abordagens *standard* estes são utilizados conjuntamente com a função objetivo para guiar a construção de novas soluções. Quando uma solução é produzida, um *standard Ant System* atualiza o rastro da seguinte forma: primeiro são enfraquecidos todos os rastros; depois os rastros que correspondem aos componentes que foram usados para construir a solução são reforçados, tendo em consideração a qualidade da solução (Taillard *et al.*, 2001).

Em termos da aplicação desta meta-heurística, muita literatura pode ser encontrada. A título de exemplo podem referir-se Gendreau, Laporte, & Potvin (2002) que a indicam como uma solução para o CVRP e, mais recentemente, este algoritmo foi adaptado por Yu, Yang, & Xie (2011) para solucionar um VRP com múltiplos depósitos, ou seja para um MDVRP.

### **3.3.2. *Simulated Annealing***

Esta meta-heurística de pesquisa, centrada na vizinhança, ultrapassa a limitação das heurísticas de melhoria local, a rápida evolução para uma solução ótima local, aceitando ações de deterioração da solução com uma probabilidade regida por um processo estatístico denominado *temperatura*. Quanto mais alto for este parâmetro mais provável é a aceitação da ação de deterioração. A *temperatura* evolui dinamicamente durante a execução do algoritmo relativamente a um esquema de arrefecimento, primeiro favorecendo uma vasta exploração e degradações frequentes, e depois gradualmente aceitando menos degradações para intensificar a obtenção de soluções de boa qualidade (Vidal, Crainic, Gendreau, & Prins, 2012).

No âmbito deste tipo de meta-heurísticas existe também a *Deterministic Annealing*, que opera de forma similar à *Simulated Annealing*, excepto no facto de existir uma regra determinística que é utilizada para a aceitação de ações.

### 3.3.3. Algoritmos Genéticos

Um algoritmo genético (AG) é uma técnica de pesquisa aleatória global que resolve problemas através de processos de imitação observados durante a evolução natural. Este paradigma foi introduzido por Holland (1975), porém 10 anos antes já era completamente reconhecido pela comunidade investigadora. Um AG puro é um método de solução de problemas genérico que usa pouca informação heurística sobre o domínio do problema. Assim pode ser aplicado a uma grande gama de problemas pouco definidos que não se associam a métodos especializados. Basicamente, um AG envolve uma **população** de *bits*, ou **cromossomas**, onde cada cromossoma codifica a solução para uma instância particular. A evolução toma lugar através da aplicação de operadores que reproduzem fenômenos naturais aplicados na natureza (e.g., reprodução, mutação) (Gendreau *et al.*, 2002).

### 3.3.4. Tabu Search

A meta-heurística denominada *Tabu Search* (TS) associa a trajetória de procura centrada na escolha do melhor vizinho da solução incumbente, com capacidades de aprendizagem, geralmente representadas como uma memória curta, média ou longa dos elementos da solução, o que substitui ou complementa de forma significativa a aleatoriedade utilizada em outras meta-heurísticas. O método afasta-se assim do ótimo local aceitando movimentos que degradam a solução, informando que eles são “os melhores” na vizinhança explorada. Este processo de decisão é fortalecido por dois mecanismos, o primeiro tentando evitar entrar em ciclo e confiando em memórias curtas para rejeitar soluções que contêm elementos *tabu* recentemente examinados, e o segundo aceitando soluções que cumprem algum *critério de aspiração* como “a melhor solução em valor ou contendo um dado elemento da solução”. De importância central são as memórias médias e longas, utilizadas para gerir procedimentos de inflexão de trajetória significativos, conhecidos como **intensificação**, por exemplo, focando a procura em torno de soluções elite enquanto promovem elementos de alta qualidade, e a **diversificação**, por exemplo, movendo a busca para uma área menos explorada de espaço de procura, promovendo elementos menos frequentes. O desafio de balancear a diversificação e a intensificação é ainda uma questão de investigação chave na literatura (Vidal *et al.*, 2012). Mais se acrescenta que estes procedimentos, intensificação e diversificação, são comuns a todas as meta-heurísticas.

Esta técnica tem sido largamente utilizada nos últimos 20 anos na resolução de problemas de encaminhamento de veículos. Passam a citar-se algumas das mais recentes: Brandão (2011) implementou uma solução TS para um VRP com frota fixa e heterogénea; e Bolduc, Laporte, Renaud, & Boctor (2010) propuseram, utilizando a meta-heurística em questão, solucionar um SDVRP com calendários de produção e encomenda.

### **3.3.5. *Neural Network***

A meta-heurística *Neural Network* (rede neuronal) é um mecanismo de aprendizagem que ajusta gradualmente um conjunto de pesos até que uma solução aceitável seja alcançada. Segundo Cordeau *et al.* (2005) esta metodologia opera num conjunto de modelos deformáveis que são essencialmente anéis candidatos a tornarem-se em rotas de veículos. Os anéis competem pelos vértices durante um mecanismo aleatório em que a probabilidade de vinculação de um destes pontos a um anel evolui através de um processo de aprendizagem.

Os mesmos autores na mesma obra referem ainda que as redes neuronais ainda não competem com a maioria das heurísticas de resolução de VRP.

### 3.4. Aproximações multiobjectivos

Resolver um problema multiobjectivos resume-se a encontrar um conjunto de soluções denominado *fronteira de Pareto*. Como os algoritmos evolucionários trabalham com uma população de soluções, estes estão bem adaptados a este tipo de problemas. Quando estes são desenvolvidos, são tidos em conta dois propósitos: a obrigatoriedade de chegar à *fronteira de Pareto* e a de encontrar soluções ao longo dessa fronteira (Jozefowicz, Semet, & Talbi, 2002). Estes propósitos correspondem às tarefas de intensificação e diversificação anteriormente referidas.

No seguimento do referido é de salientar que as aproximações multiobjectivos são formas de resolução de VRP que surgem regularmente no mundo real. Um exemplo deste facto é caso de estudo de Gupta, Singh, & Pandey (2010) denominado *Multi Objective Fuzzy Vehicle Routing Problem with Time Windows and Capacity Constrains*, onde existem os seguintes objetivos:

- Minimização do tamanho da frota;
- Minimização do total de distância viajada;
- Maximização da classificação média de satisfação dos utilizadores;
- Minimização do tempo de espera por veículos.

Tendo em mente os objetivos, as restrições do problema são:

- O tempo de serviço para cada cliente é o tempo considerado tolerável por cada um.
- Cada veículo deve ser atribuído aos clientes de forma a não ultrapassar a sua capacidade.
- Cada cliente é servido apenas por um veículo.
- Para cada cliente há apenas dois pontos geográficos (clientes) diretamente ligados a ele: o ponto a partir do qual é possível chegar diretamente até ele e aquele para o qual é possível viajar diretamente (com um veículo).
- Os clientes são servidos dentro de uma dada janela temporal.

Como se pode começar a vislumbrar pelo enunciado, o caso real da problemática prende-se com o serviço de autocarros escolares (no caso de estudo entre os vários departamentos dispersos de uma Universidade e o resto da cidade).

Para a resolução desta problemática foi adaptada e utilizada a meta-heurística de Algoritmos Genéticos.

## **4. Sistema de Informação Geográfica de apoio à decisão**

---

Este capítulo marcará a transição entre a investigação e a ação e apresentará a implementação do trabalho de ação propriamente dito. Começar-se-á com uma investigação sumária sobre os SIG, suas características e componentes e com a apresentação das mais-valias da sua utilização para apoio à decisão em VRP. De seguida passar-se-á à descrição da problemática a resolver, designadamente assumindo os pressupostos e as funcionalidades consideradas necessárias, os tipos de VRP a trabalhar e os algoritmos de resolução a aplicar. Antes de se iniciar a apresentação dos elementos da aplicação propriamente ditos, fundamentar-se-ão as tecnologias de desenvolvimento de *software* apropriadas para o caso e apresentar-se-á também o estudo e construção do Sistema de Informação que formará o suporte do conhecimento que a aplicação manipulará. Por fim avançar-se-á para a apresentação das funcionalidades da aplicação, começando pelas de povoamento do sistema de informação ou, por outras palavras, de caracterização do VRP, e avançando para as de apoio à decisão. Esta exposição demonstrativa da aplicação será documentada com imagens e esclarecimentos sobre cada funcionalidade desenvolvida.

#### 4.1. Enquadramento

Apesar de os Sistemas de Informação Geográfica (SIG) terem ganho grande força com o recente choque tecnológico de electrónica, telecomunicações e informática, é possível encontrar-se definições que remontam a sensivelmente duas décadas atrás. Por se considerar que estas se mantêm atuais, é de relevo mencionar algumas. Assim, um SIG pode ser definido como sendo:

- um sistema para captura, armazenamento, verificação, manipulação, análise e apresentação de informação que está espacialmente referenciada na Terra (DoE, 1987).
- qualquer conjunto de procedimentos manual ou computadorizado utilizado para armazenar e manipular informação geograficamente referenciada (Aronoff, 1989).
- uma tecnologia de informação que armazena, analisa e mostra informação espacial e não espacial (Parker, 1988).
- um caso especial de sistemas de informação onde a base de dados consiste de observações sobre características, atividades ou eventos, que são definidos em pontos, linhas ou áreas espacialmente distribuídas. Um SIG manipula informação sobre estes pontos, linhas e áreas, para devolver informação mediante consultas e para a realização de análise (Dueker, 1979).

Resumindo e direcionado a definição de SIG para a problemática alvo desta dissertação, é válido apresentar este instrumento como sendo “uma ferramenta que permite manipular dados georreferenciados e alfanuméricos para, a partir de análises espaciais, apoiar a tomada de decisão espacial, como a definição do melhor roteiro de entregas a ser seguido” (Lima, Lima, & Silva, 2012, p. 181). No seguimento e completando o conceito anterior, associando-o à sua utilidade no apoio à tomada de decisão, Malczewski (1999) transmite que os SIG e a tomada de decisão multicritério (TDMC) podem beneficiar entre si. Por um lado, as técnicas e procedimentos SIG têm um papel importante na análise de problemas de TDMC, oferecendo capacidades únicas de automatização, gestão e de análise de uma grande variedade de dados geográficos para a tomada de decisões. Por outro lado, as TDMC têm a si associadas um conjunto amplo de metodologias que oferecem uma coleção de técnicas e procedimentos passíveis de serem incorporados em SIG baseados em, e para tomadas de decisão.

## 4.2. Características

As definições mencionadas anteriormente indiciam a multiplicidade de usos interdisciplinares atribuíveis aos SIG. Com base em Davis & Neto (2001), passam-se a apontar agora as suas funcionalidades e características gerais:

- Inserção e manutenção, em base de dados, de informações espaciais provenientes de dados cartográficos, imagens de satélite, etc;
- Oferecimento de mecanismos para combinar as várias informações, através de algoritmos de manipulação e análise, bem como para consultar, recuperar e visualizar os conteúdos georreferenciados da base de dados.

As características descritas apontam para que um SIG seja composto por uma base de dados povoada com informações georreferenciadas e uma aplicação que permita a gestão das mesmas. No entanto e voltando à ideia de Malczewski, os SIG que motivam este trabalho deverão ter o valor acrescentado de incorporar algoritmos/mecanismos de apoio à decisão que, no caso dos VRP, auxiliem na definição das rotas ótimas tendo em conta os objetivos e as restrições de cada caso. Além disso, é também objetivo deste trabalho o uso de tecnologias *Web* que facilitem a inserção, manipulação, visualização e análise de dados georreferenciados, tornando as operações de carregamento destes praticamente transparentes ao utilizador.

## 4.3. Componentes

Tendo em mente as características enunciadas anteriormente, estas podem ser decompostas nos componentes gerais comuns de SIG:

1. **Base de Dados:** suporte da informação textual e georreferenciada.
2. **BackOffice:** Aplicação informática de gestão da informação que povoa a Base de Dados: inserção, edição e remoção de dados.
3. **FrontOffice:** Aplicação informática de interface com o utilizador/decisor permitindo a consulta e apresentação de informações textuais e georreferenciadas constantes na base de dados, envolvendo as funções algorítmicas que permitirão a análise e construção espacial de informação (nomeadamente composição de rotas).



Questões de confidencialidade, de permissão de acesso à informação e de competências/atribuições profissionais poderão ditar a arquitetura do *Backoffice* e do *FrontOffice* a implementar, podendo estas duas componentes do SIG estar embebidas numa só aplicação.

#### **4.4. A ferramenta SIG desenvolvida**

Tendo em mente as definições, características e componentes de um SIG, fica-se com uma visão clara da ligação que é possível estabelecer entre esta ferramenta e o apoio à decisão no caso dos VRP. Uma ferramenta que permita agregar dados textuais e georreferenciados sobre clientes, depósitos, veículos e encomendas (poderiam ser referidos outros, mas estes são os elementos essenciais), e tenha aptidões para verificar as distâncias entre estes, mostrá-los num mapa e desenhar automaticamente rotas entre os mesmos, tem grande potencial no apoio à decisão. Com estas características, se a ferramenta for dotada de algoritmos de otimização de rotas que forneçam ao gestor soluções para os VRP com que se depara, então este terá em mãos um importante elemento de trabalho, que lhe dará grande apoio às tomadas de decisão no âmbito em causa.

Passam a enunciar-se os pressupostos e funcionalidades consideradas de relevo para que a ferramenta a desenvolver seja o instrumento de apoio à decisão ambicionado.

##### **4.4.1. Pressupostos e funcionalidades**

Após a investigação descrita nos capítulos anteriores sobre VRP, suas características, objetivos, instâncias e formas de resolução, assume-se como tendo grande interesse o desenvolvimento de um SIG genérico que apoie a decisão na definição de rotas de um amplo leque de empresas. Tal sistema, cujos alicerces devem ser formados pelo conhecimento resultante da investigação sobre VRP, deve oferecer as seguintes funcionalidades:

- Gestão de informação textual e georreferenciada de depósitos e clientes (pontos geográficos)
- Gestão de frotas
- Gestão de encomendas
- Apoio à decisão:
  - Percurso entre dois pontos geográficos relevantes (depósitos ou clientes)
  - Definição de rotas personalizadas
  - Definição de rotas automáticas
  - Gestão (visualização e remoção) de rotas

- Visualização de histórico ou comparação de resultados obtidos com a definição de diferentes tipos de rotas

A gestão de informação georreferenciada de depósitos e clientes está relacionada com a inserção e manipulação de informações textuais e espaciais sobre estas entidades num suporte de informação, tipicamente numa base de dados. A gestão de frotas prende-se com a inserção de informação sobre os veículos disponíveis e com a vinculação dos mesmos a depósitos. Quanto à funcionalidade de gestão de encomendas, esta deverá permitir a inserção de informações chave para a definição do problema a resolver, como o peso da encomenda e a data de entrega. Além disso, como é óbvio, cada encomenda encontrar-se-á ligada a um cliente e, assim, a um ponto geográfico. As funcionalidades que se descreveram até aqui poderão englobar-se no que anteriormente se chamou de *BackOffice*.

No que às funcionalidades de *FrontOffice* diz respeito, estas serão todas as que apoiem a decisão, baseadas nas informações submetidas no *BackOffice*. Um dos pontos-chave para a definição das rotas é a distância entre os pontos geográficos a visitar (depósitos e clientes). Assim sendo, uma das funcionalidades básicas mas imprescindível no sistema que se pretende implementar é a visualização em mapa interativo do percurso entre dois pontos, a respetiva distância e o tempo de viagem. Além disso, convém dar ao decisor humano a possibilidade de definir ele próprio as rotas manualmente, escolhendo o veículo e as encomendas que quer atribuir a cada rota. Porém, dado que o problema em causa é um VRP, a funcionalidade chave do mesmo será o cálculo automático e otimizado de rotas, segundo um ou mais algoritmos dos que se abordaram nos capítulos de resolução de VRP. Assim, o sistema definirá automaticamente as rotas otimizadas dos veículos da frota para um dado dia, tendo em conta as encomendas existentes e as capacidades dos referidos veículos. Porém, tendo em conta exigências reais do mundo empresarial e outras premissas, como por exemplo o maior conhecimento de uma zona geográfica por parte de dado condutor (associado a um veículo), é essencial que a ferramenta permita a gestão (visualização e remoção) de rotas, assim como a definição manual de rotas anteriormente referida. Este conjunto de funcionalidades dotará a aplicação de uma maior maleabilidade.

Para concluir esta análise prévia às funcionalidades consideradas indispensáveis para o SIG a desenvolver, resta mencionar que se considera importante, dado que a ferramenta é de apoio à decisão, a incorporação de informações de histórico ou estatísticas sobre rotas anteriormente geradas.

#### **4.4.2. O VRP**

Após a descrição prática dos pressupostos e funcionalidades, avança-se para a descrição dos tipos de VRP que se pretendem tratar na aplicação a desenvolver, assim como quais os algoritmos a utilizar para a sua resolução.

##### **4.4.2.1. Instâncias a resolver**

Conforme anteriormente referido pretende-se que o SIG a desenvolver seja genérico para que apoie a decisão na definição de rotas de um vasto leque de empresas. Tendo este pressuposto como referência e após reflexão sobre quais os casos de VRP que ocorrem mais frequentemente, passam a descrever-se as características dos problemas que se desejam tratar, ligando-as às instâncias de VRP anteriormente analisadas:

- CVRP:
  - Veículos com capacidades de carga finitas
- MDVRP:
  - Existência de vários depósitos
  - Objetivo de minimização das distâncias percorridas por toda a frota

Explicitando melhor as opções tomadas, é possível afirmar-se que a seleção do CVRP (*Capacitated Vehicle Routing Problem*) é natural, dado ser o caso mais frequente de VRP, uma vez que em aplicações reais existe a restrição óbvia dos veículos em termos de capacidade. A opção de tratar os MDVRP (*Multiple Depot Vehicle Routing Problem*) surge com o intuito de alargar o espectro de problemas tratados pela aplicação, considerando-se ser esta uma evolução que, apesar de natural, aumenta largamente o número de problemas suportados pela aplicação, tornando-a numa ferramenta viável.

##### **4.4.2.2. Soluções a aplicar**

Em termos de resolução do problema, na funcionalidade anteriormente referida como sendo de geração automática de rotas otimizadas, considera-se pertinente oferecer ao utilizador da aplicação a possibilidade de utilização de mais do que um algoritmo para que seja possível a comparação de resultados. Assim sendo, e lembrando as formas de resolução de VRP, será

interessante utilizar algoritmos de tipos diferentes, como por exemplo um do tipo *Cluster-first, route-second* e outro que, baseado numa lógica de inicialização, crie as rotas para os veículos apenas atendendo à restrição de capacidade dos mesmos, como por exemplo o algoritmo do vizinho mais próximo.

Após a utilização de qualquer uma das referidas heurísticas construtivas referidas, será também academicamente interessante possibilitar a utilização de um método de melhoria local, que verifique se é possível aperfeiçoar as rotas criadas, para um ótimo local.

As heurísticas construtivas utilizadas e o método de melhoria serão explicados pormenorizadamente aquando da apresentação das formas de criação automática de rotas na aplicação.

#### **4.4.3. Tecnologias**

Tendo em conta o carácter da aplicação a desenvolver, considerou-se vantajoso que esta esteja acessível ao seu utilizador, por exemplo o gestor da frota de uma empresa, não apenas na máquina onde esteja instalada, mas em qualquer ponto onde se encontre. Assim sendo, direccionaram-se os trabalhos para o desenvolvimento de um Sistema Web de Informação Geográfica de apoio à decisão, no âmbito de problemas de encaminhamento de veículos.

Para o desenvolvimento do SIG descrito pelas premissas do subcapítulo anterior, decidiu-se que a utilização de tecnologias *Open Source* seria favorável. A filosofia inerente e o carácter gratuito em termos de licenças de utilização e desenvolvimento são consideradas mais-valias para o programador e para qualquer empresa cliente.

Deste modo estipulou-se o uso das ferramentas:

- **MySQL** - Sistema Gestor de Base de Dados.
- **phpMyAdmin** - Interface gráfica facilitadora da gestão da Base de Dados MySQL.
- **Apache** - Servidor Web local, elemento imprescindível à construção de um *website* dinâmico (que usa Bases de Dados).
- **HTML + PHP + JavaScript** - Linguagens de *scripting* para:
  - Desenho e estruturação da aplicação;
  - Comunicação entre a aplicação e a base de dados;
  - Comunicação entre a aplicação e o Google Maps.

- **Google Maps API V3** – Conjunto de serviços baseados na Web que permitem obter informações sobre distâncias, tempos de percurso e percursos rodoviários, assim como importar mapas para uma aplicação própria e desenhar rotas nos mesmos.

#### **4.4.4. Suporte de informação**

O sistema de informação de suporte à aplicação foi desenhado a partir das especificações descritas em 4.4.1., que se baseiam nos elementos dos VRP, apresentados em 2.2.. A Figura 4 apresenta o Diagrama de Entidades e Relações (DER) descritivo do mesmo.

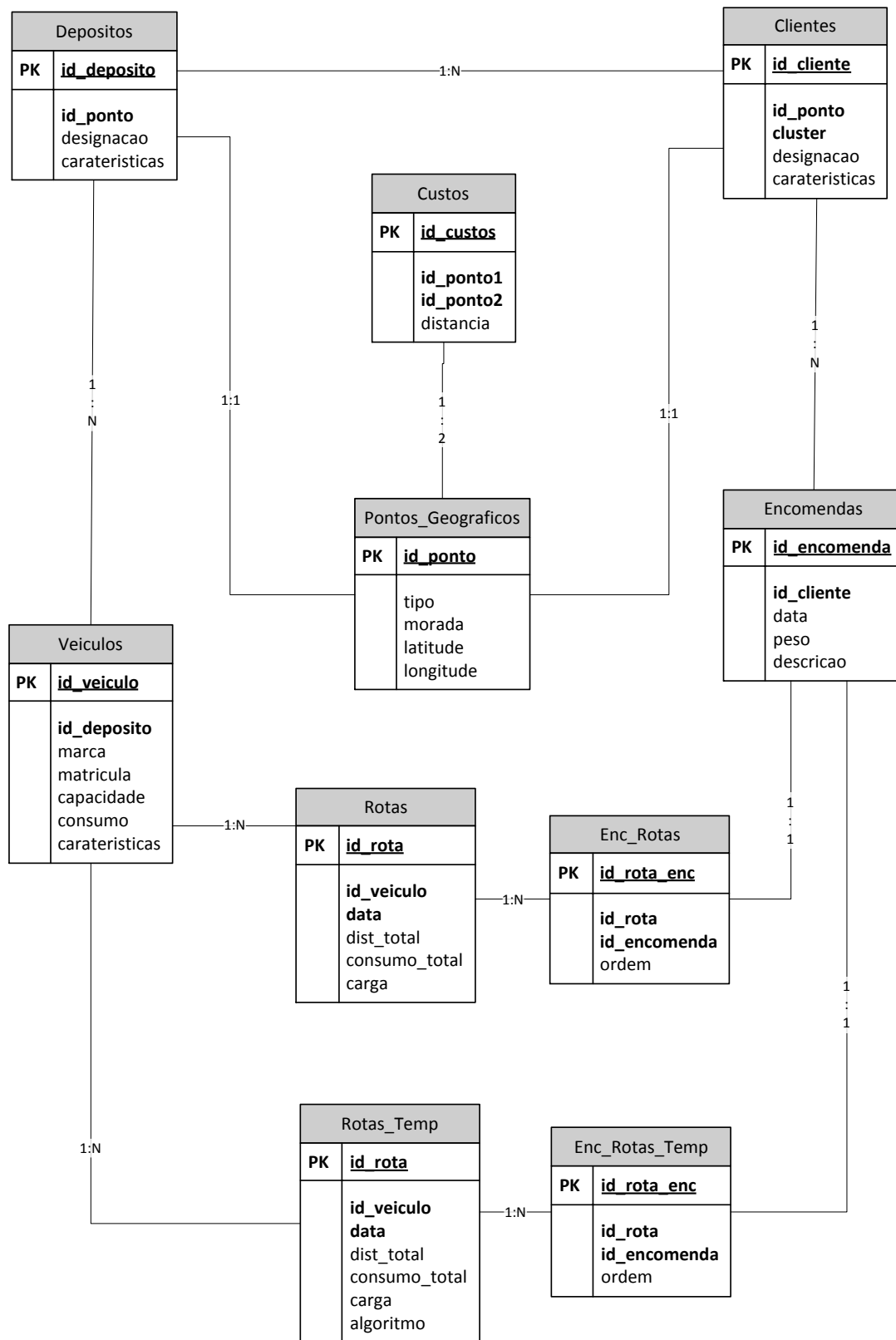


Figura 4 – Diagrama de Entidades e Relações para sistema de informação

O DER apresentado mostra, além das entidades e relações entre as mesmas, os campos que se considera imprescindível guardar na base de dados. Passa-se a descrever sucintamente os pormenores de maior relevo do diagrama.

### **Pontos Geográficos**

A tabela central e base de todas as outras é a *Pontos\_Geograficos*. Aqui são armazenadas as informações georreferenciadas sobre clientes e depósitos, que possibilitarão a sua marcação no mapa e a verificação das distâncias entre eles. É de salientar o campo *tipo*, existente nesta tabela, que indicará se dado ponto é um cliente ou um depósito.

### **Clientes e Depósitos**

As tabelas *Clientes* e *Depositos*, apesar de terem campos semelhantes e relações de 1 para 1 com a tabela *Pontos\_Geograficos*, foram separadas por razões de inteligibilidade do sistema de informação, uma vez que uma estará relacionada com os veículos e outra com as encomendas. Além disso, repare-se no campo *cluster*, que assinala a diferença em termos de estrutura entre as tabelas. Este campo foi introduzido para possibilitar a utilização de algoritmos de resolução de VRP que agrupem os clientes em *clusters* (do tipo *Cluster-first Route-second*). No caso, a cada depósito corresponderá um *cluster* e os clientes pertencentes a esse grupo são os que têm o depósito em questão como o mais próximo. A definição destes *clusters* será melhor explicada posteriormente aquando da apresentação da gestão dos clientes no sistema.

### **Custos**

Esta tabela, com uma relação de 1 para 2 com a tabela *Pontos\_Geograficos*, foi concebida para albergar os custos de deslocação entre dois pontos geográficos. No caso apenas se albergará a distância entre dois pontos mas, sem grande problema se acrescentarão futuramente outros campos como tempos de percurso, ou preço de portagens entre dois pontos.

### **Encomendas**

A importante tabela *Encomendas* alberga as informações sobre uma entidade chave para a construção das rotas, como a data e o peso das encomendas. A data é importante para identificar que encomendas precisam ser servidas em cada dia e o peso da encomenda é utilizado para a avaliação de que encomendas cabem em cada veículo. Outros campos, como o



volume das encomendas, poderiam ser considerados porém, porque o procedimento de construção de rotas seria semelhante, basta a restrição de peso para a afetação das encomendas às rotas de cada veículo. Como se pode verificar, esta tabela contém uma ligação de N para 1 com os clientes, representando que cada cliente pode ter várias encomendas.

### **Veículos**

Outra tabela chave para a construção das rotas é a tabela *Veiculos*, uma vez que alberga informações sobre uma entidade que, em conjunto com a data, identificará univocamente cada rota (definiu-se que para cada dia, cada veículo apenas realizaria uma rota). O principal campo desta tabela é *capacidade* que constituirá a restrição que condicionará o número de encomendas que o veículo pode levar em cada rota. Além deste campo destaca-se o campo *consumo*, que será utilizado para estimar o consumo médio do veículo para cada rota e o consumo total para cada dia (do conjunto de todas as rotas).

### **Rotas e Enc\_Rotas**

As tabelas *Rotas* e *Enc\_Rotas* albergam respetivamente informações globais sobre cada rota e informações específicas sobre cada ponto a visitar em cada rota. Na tabela *Rotas* o sistema alberga a data e o veículo que definem a rota, assim como informações como a distância total, o consumo total e a carga (peso de todas as encomendas) que caracterizam a mesma. A tabela *Enc\_Rotas* estabelece a ligação entre as tabelas *Rotas* e *Encomendas*, caracterizando as encomendas que serão servidas em cada rota. Além disso, esta tabela incorpora o campo *ordem* que indica a ordem de serviço das encomendas na rota.

### **Rotas\_Temp e Enc\_Rotas\_Temp**

As tabelas *Rotas\_Temp* e *Enc\_Rotas\_Temp* albergam informações semelhantes às das tabelas *Rotas* e *Enc\_Rotas*, com a diferença de estas serem temporárias, ou seja, ainda estão em análise por parte do decisor humano. Tendo as mesmas características das duas últimas que se analisaram, estas são necessárias para que seja possível ao sistema gerar e apresentar as rotas, com a utilização de algoritmos de resolução de VRP. Observando atentamente *Rotas\_Temp*, repara-se que esta difere de *Rotas* exatamente por um campo *algoritmo* que guarda a informação sobre que algoritmo foi utilizado para construir cada rota. Após a análise das rotas geradas, se o utilizador considerar que as mesmas lhe agradam, poderá guardá-las.

Esta ação fará com que o sistema copie os registos respetivos de *Rotas\_Temp* e *Enc\_Rotas\_Temp* para *Rotas* e *Enc\_Rotas*.

#### 4.4.5. A aplicação

Este subcapítulo marca o final da investigação e o início da ação, ou seja, a partir daqui apresentar-se-á o resultado do estudo, análise e planeamento realizados: a implementação do SIG de apoio à decisão para VRP. Começar-se-á por se apresentar as funcionalidades de *BackOffice*, ou seja, de povoamento e gestão do sistema de informação ou, por outras palavras, as funcionalidades que criam o VRP. De seguida, apresentar-se-ão as funcionalidades de apoio à decisão (*FrontOffice*), ou seja, as que permitem a resolução do VRP.

Na aplicação desenvolvida, cuja página de entrada se apresenta abaixo, é possível identificar perfeitamente a zona de *BackOffice*, de gestão da informação textual e georreferenciada constante na base de dados, e a zona de *FrontOffice*, com funcionalidades de apoio à decisão.

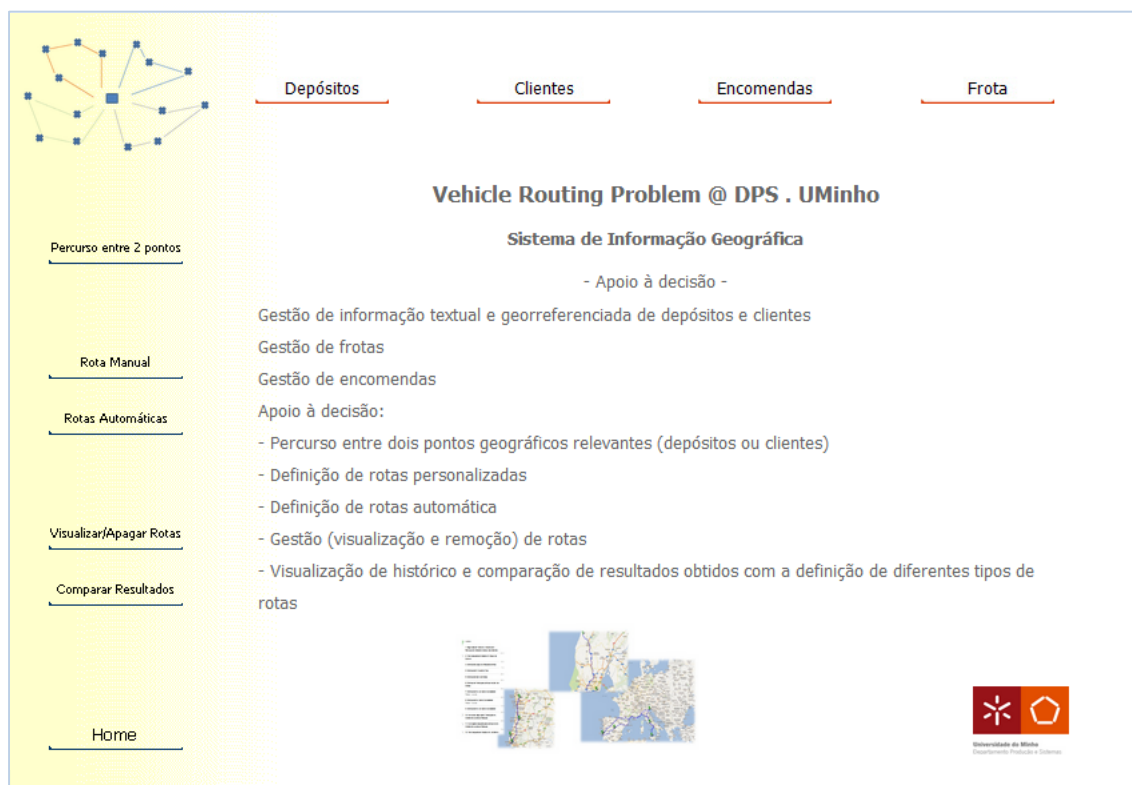


Figura 5 – Página inicial do SIG de apoio à decisão

Passando à análise da área de *BackOffice*, caracterizada pelas hiperligações *Depósitos*, *Clientes*, *Encomendas* e *Frota*, é de salientar que esta ordem não foi estabelecida ao acaso. A mesma foi definida consoante as precedências de existência de registos das entidades na base de dados que se passam a explicar.

A entidade *Depósitos* não depende de nenhuma outra, ou seja, é possível, por exemplo, inserir um registo de um depósito sem que existam registos de clientes, veículos ou encomendas no sistema de informação. Assim sendo, é lógico que seja apresentada em primeiro lugar.

A entidade *Clientes*, apesar de não depender de nenhuma outra, encontra-se depois de *Depósitos* porque se não existirem registos de depósitos não é lógico que existam clientes. Além disso, o cálculo do *cluster* de cada cliente, definido pelo depósito mais próximo do mesmo, teoricamente impõe a existência de registos de depósitos em base de dados. A palavra “teoricamente” da frase anterior tem o intuito de fazer entender que, para a inserção de novos clientes e associação dos mesmos a um *cluster*, seria suposto que existissem registos de depósitos. Porém tal não é obrigatório uma vez que, na aplicação desenvolvida, aquando da inserção de novos depósitos os *clusters* de todos os clientes são recalculados, dado que o depósito mais próximo de cada um pode ter sido alterado.

A entidade *Encomendas*, como é óbvio, dependerá da entidade *Clientes*, uma vez que cada encomenda criada será associada ao registo de um cliente.

Por fim, a entidade *Frota*, que corresponderá ao conjunto dos veículos disponíveis, dependerá da entidade *Depósitos*, dado que o registo de cada veículo estará obrigatoriamente associado a um depósito.

Cada hiperligação referente a cada uma das referidas entidades permitirá ao utilizador a inserção de novos registos, a listagem dos registos existentes consoante vários critérios, a edição dos dados referentes a cada registo e a eliminação de registos. Estas funcionalidades serão analisadas em *4.4.5.1. Povoamento e gestão do sistema de informação*.

Para a disposição gráfica das funcionalidades do *FrontOffice*, *Percurso entre 2 pontos*, *Rota Manual*, *Rotas Automáticas*, *Visualizar/Apagar Rotas* e *Comparar Resultados*, foi aplicada uma lógica semelhante, mas não tão evidente. A prioridade neste caso, não teve tanto a ver com dependências, mas com uma possível ordem normal de utilização. Considera-se normal que se queira saber o percurso entre dois pontos antes de avançar para a construção de rotas completas. Também é defensável que, antes de se gerarem todas as rotas automaticamente, se

ensaie a construção das mesmas consoante as opções do utilizador. Depois da construção das rotas por parte do utilizador, o mesmo poderá querer verificar se as rotas automáticas, ou otimizadas tendo em conta os algoritmos utilizados para a sua construção, melhoram as suas opções. Apesar desta forma de utilização pensada, a oposta também seria defensável (primeiro construir as rotas automaticamente e utilizar de seguida a funcionalidade de rotas manuais para “retocar” as rotas que os algoritmos tivessem construído de forma desapropriada). Após a construção de rotas surgem logicamente as funcionalidades de visualizar e de apagar as rotas geradas. Concluindo as funcionalidades de apoio à decisão, surge em último lugar a de comparação de resultados que, além do que o nome descreve, permitirá ao utilizador ter alguma percepção do histórico de resultados obtidos por rotas anteriormente traçadas.

Seguidamente apresentar-se-ão as funcionalidades do *BackOffice*, de povoamento e gestão do sistema de informação, acompanhadas das imagens respetivas da aplicação.

#### **4.4.5.1. Povoamento e gestão do sistema de informação**

Neste ponto, tal como anteriormente referido, apresentar-se-ão as funcionalidades do *BackOffice*, mas não exaustivamente, ou seja, não serão apresentadas todas as operações de inserção, visualização, listagem, edição e remoção de registos. Serão apenas apresentados os exemplos considerados pertinentes de forma a reduzir a redundância e serão salientados pormenores como o da formação e manutenção da tabela (matriz) de custos e dos *clusters*, e as validações de dados realizadas.

##### **4.4.5.1.1. Ecrãs de gestão**

Após o clique na respetiva hiperligação, cada entidade conterà um ecrã que permitirá a inserção, visualização, listagem, edição e remoção de registos, que se denominou de ecrã de gestão. Na Figura 6 apresenta-se, a título de exemplo, o ecrã de gestão da entidade *Encomendas*.



Figura 6 – Ecrã de gestão da entidade *Encomendas*

Apesar de as opções gerais (inserção, visualização, listagem, edição e remoção de registos) serem as mesmas para todas as entidades, este ecrã terá pequenas diferenças entre elas, principalmente ao nível das opções de escolha para listagem de registos.

Abordar-se-ão de seguida as funcionalidades de gestão do sistema, assim como as restrições e validações inerentes a cada.

#### 4.4.5.1.2. Inserção de registos, a matriz de distâncias e os *clusters*

A inserção de registos em cada entidade partirá do ecrã de gestão de cada uma, da opção *Novo Depósito*, *Novo Cliente*, *Nova Encomenda* ou *Novo Veículo*. Apesar da similaridade aparente, estas opções serão bastante diferentes porque, como se verá adiante, é com a inserção de registos que se povoa a matriz distâncias (tabela *Custos*, que guardará as distâncias entre os pontos geográficos) e se definem os *clusters*. Assim sendo, este é um passo de grande importância para a criação e resolução dos VRP.

Começando pelo que as opções referidas têm em comum, a inserção de registos nas tabelas referentes às entidades, estas são divididas em duas partes: a inserção da informação e a validação e submissão da mesma. Caso a informação submetida não passe nos testes de validação, o registo não será criado, sendo o utilizador informado desse desfecho.

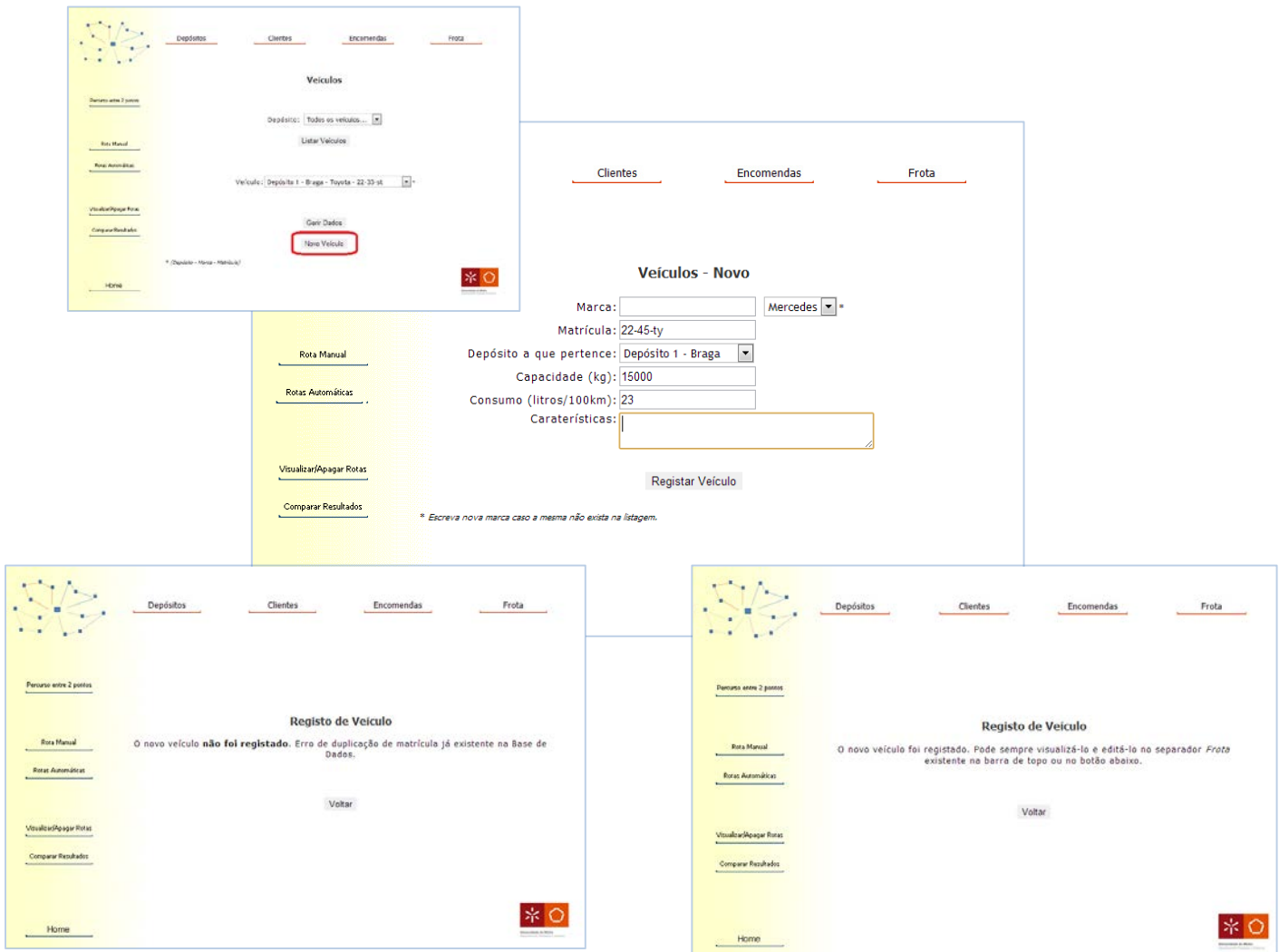


Figura 7 – Conjunto de ecrãs exemplificativos da operação de inserção de registo

As validações existentes aquando da submissão de um novo registo são:

- Depósitos: Designação do depósito inexistente na base de dados e a morada é reconhecida pelo *Google Maps*;
- Cientes: Designação do cliente inexistente na base de dados e a morada é reconhecida pelo *Google Maps*;

- Encomendas: Sem validações necessárias porque os clientes e datas do formulário são escolhidos utilizando *combo-boxes*, de entre valores válidos (clientes existentes ou datas válidas);
- Veículos (frota): Matrícula do veículo inexistente na base de dados.

Porém, conforme anteriormente referido, a inserção de registos está intimamente relacionada com todo o processo de resolução do VRP. Aliado ao imprescindível facto de ser neste momento que se inserem as moradas dos clientes e dos depósitos, está a situação de ser aqui que se calculam as distâncias entre estes pontos geográficos e se registam no sistema, sendo que esta operação é transparente ao utilizador, que apenas insere a morada. Assim, sempre que é inserido um novo ponto geográfico (depósito ou cliente), após a validação da morada através de uma chamada ao *Google Maps*, é calculada e registada a distância entre esse ponto e todos os outros (também através de invocações ao *Google Maps*), criando-se assim uma linha da importante matriz distâncias. Como se infere na afirmação anterior, assume-se que a distância de percurso em sentido contrário, entre os outros pontos e o novo ponto, é a mesma. Obtém-se então uma matriz triangular que, apesar de ter como defeito não representar com total exatidão as distâncias rodoviárias de ida e de volta (assumem-se iguais), tem como vantagem a poupança de espaço de armazenamento de registos em base de dados e a velocidade de execução dos algoritmos na procura das distâncias entre os pontos, uma vez que o número de registos será metade.

Outro aspeto de relevo, para a problemática alvo desta dissertação, no âmbito da inserção de novos registos é a criação de *clusters*. Lembra-se que cada cliente deve pertencer ao *cluster* identificado pelo depósito mais próximo de si. Relativamente a este aspeto é de salientar o comportamento do sistema com a inserção de novos clientes e o seu comportamento com a inserção de novos depósitos. Quando se insere um novo cliente, o sistema irá procurar qual o depósito mais próximo de si, anexando o cliente a esse *cluster* (colocando o identificador (*id*) do depósito no campo *cluster* do registo do cliente). Quando se insere um novo depósito, dado que este procedimento pode alterar os *clusters* de vários clientes, desencadeia-se o processo de novo cálculo e registo de *clusters* para todos os clientes, tendo já o novo depósito em consideração. Apesar de esta operação parecer (e ser) computacionalmente custosa, é imprescindível e não será realizada muitas vezes, dado que a inserção de um novo depósito é um evento raro.

#### 4.4.5.1.3. Visualização, edição e eliminação de registos

Para a visualização, edição e eliminação de registos, considerou-se que seria bastante intuitivo e manteria a utilização da ferramenta num patamar de simplicidade considerável se se utilizasse o mesmo formulário para as três funcionalidades. Repare-se na Figura 8, na solução implementada.

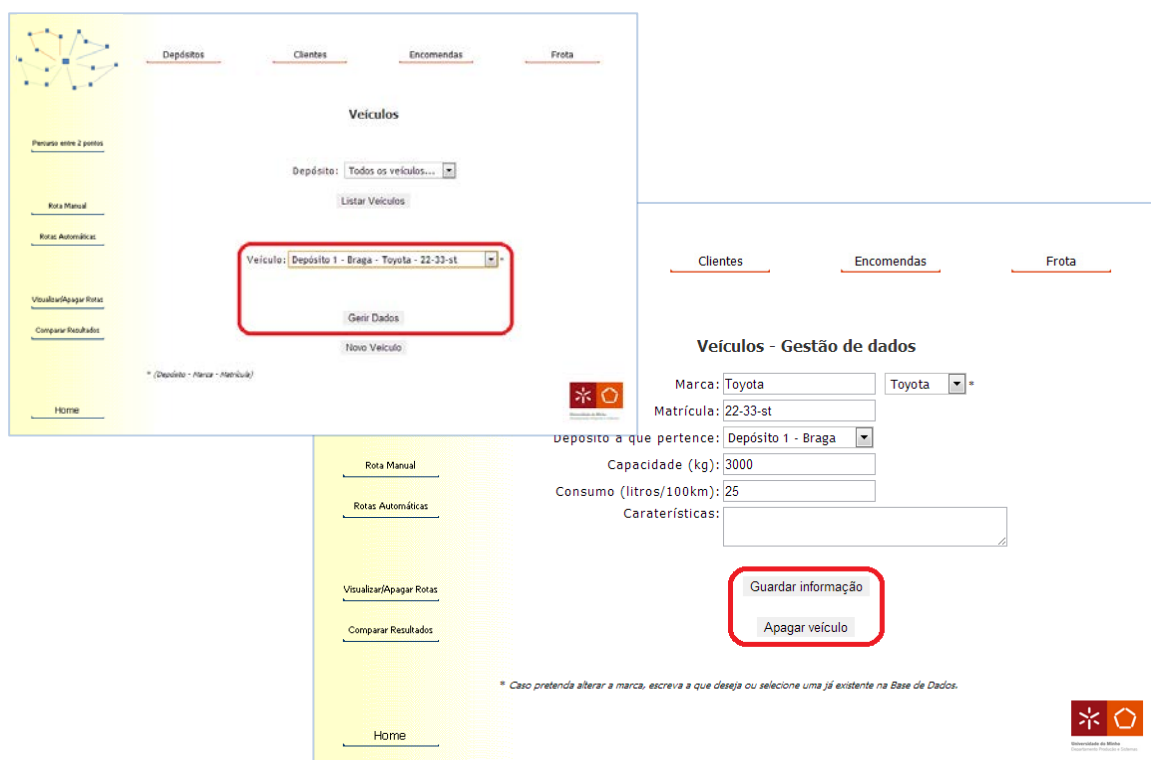


Figura 8 – Conjunto de ecrãs exemplificativos das operações de visualização, edição e eliminação de registo

Como se pode ver, as três funcionalidades em questão partem do identificador unívoco do registo (depósito, marca e matrícula no caso dos veículos) e avançam para um formulário que permite a execução das mesmas, ou seja, permite a visualização, a alteração ou edição dos dados do registo e a eliminação do mesmo. Apesar destas funcionalidades utilizarem os mesmos ecrãs e formulários, tanto a edição como a eliminação de registos têm características que importa salientar.

Da edição de registos é de valor referir que no caso dos pontos geográficos (depósitos e clientes), não é possível alterar as informações sobre a morada. Esta decisão deve-se ao peso que se considera que esta alteração tem ao nível de validações de morada, cálculo de distâncias



e cálculo de *clusters versus* a frequência real com que um cliente ou depósito mudam de morada. A solução que o sistema oferece para o caso de um cliente ou depósito mudar de morada é a eliminação do registo e a criação de novo.

Sobre a eliminação de registos, convém referir que esta ação despoleta eliminações lógicas em cascada, nomeadamente com a eliminação de:

- Depósito: eliminam-se todos os veículos desse depósito e rotas que esses veículos tinham a si associadas, ficando as encomendas livres para ser atribuídas a outras rotas.
- Cliente: eliminam-se todas as encomendas desse cliente, inclusivamente das rotas onde já estavam inseridas.
- Encomenda: elimina-se o registo da encomenda na rota onde estava inserida.
- Veículo: eliminam-se as rotas que o veículo realizou ou realizaria, ficando as encomendas em questão livres para ser atribuídas a outras rotas.

#### **4.4.5.1.4. Listagem de registos**

As listagens de registos permitem ao gestor gerar catálogos de registos de entidades que tenham algo em comum.


Listagens permitidas no sistema:

- Todos os clientes;
- Listar clientes por morada total ou parcial, através da inserção de uma palavra ou conjunto de palavras contidas no campo *morada*;
- Todas as encomendas;
- Todas as encomendas de dada data;
- Todas as encomendas de dado cliente;
- As encomendas de dado cliente para dada data;
- Todos os veículos;
- Todos os veículos de dado depósito;

As listagens disponibilizadas, apesar de conterem campos que são introduzidos pelo utilizador, não precisam de dispositivos de validação porque não criam inconsistências na base de dados nem no sistema. O único campo que poderia causar esses problemas seria a *data* nas encomendas, porém essa situação foi ultrapassada com a utilização de uma *combobox* que

apenas oferece ao utilizador datas válidas e com encomendas registadas (as outras não interessariam pois apenas retornariam listagens vazias).

Na Figura 9 podem visualizar-se exemplos de listagens nomeadamente: todos os clientes que contenham a letra *b* na morada; todas as encomendas do dia *01-01-2013*, e todos os veículos de *Depósito 2 – Coimbra*.




### Cientes - Listagem

Designação	Morada	Caraterísticas
Cliente 7 - Barcelos	Barcelos, Portugal	
Cliente 16 - Louisa, Coimbra	Louisa, Coimbra, Portugal	
Cliente 23 - Sesimbra	Sesimbra, Portugal	
Cliente 32 - Braganca	Braganca, Portugal	
Cliente 33 - Ponte da Barca	Ponte da Barca, Portugal	
Cliente 35 - Castelo Branco	Castelo Branco, Portugal	
Cliente 47 - Barrancos	Barrancos, Portugal	


[Voltar](#)


[Home](#)



### Encomendas - Listagem


Cliente	Data	Peso (kg)	Descrição
Cliente 31 - Chaves	2013-01-01	600	
Cliente 32 - Braganca	2013-01-01	600	
Cliente 33 - Ponte da Barca	2013-01-01	600	
Cliente 34 - Guarda	2013-01-01	600	
Cliente 35 - Castelo Branco	2013-01-01	600	
Cliente 36 - Portalegre	2013-01-01	600	
Cliente 37 - Serpa	2013-01-01	600	
Cliente 38 - Tavira	2013-01-01	600	
Cliente 39 - Portimao	2013-01-01	600	
Cliente 40 - Sagres	2013-01-01	600	
Cliente 41 - Odeceixe	2013-01-01	600	
Cliente 42 - Evora	2013-01-01	600	
Cliente 43 - Lamego	2013-01-01	600	
Cliente 44 - Vila Real	2013-01-01	600	
Cliente 45 - Viana do Castelo	2013-01-01	600	
Cliente 46 - Espinho	2013-01-01	600	
Cliente 47 - Barrancos	2013-01-01	600	
Cliente 48 - Mertola	2013-01-01	600	
Cliente 49 - Resende	2013-01-01	600	
Cliente 50 - Vigo	2013-01-01	600	





### Veículos - Listagem

Depósito	Matrícula	Marca	Capacidade (kg)	Consumo (litros/100km)	Caraterísticas
Depósito 2 - Coimbra	34-23-ed	Mercedes	3500	30	
Depósito 2 - Coimbra	54-43-er	Toyota	3000	25	
Depósito 2 - Coimbra	43-76-dc	Toyota	3000	25	



[Voltar](#)

[Home](#)

Figura 9 – Exemplos de listagens

#### 4.4.5.2. Funcionalidades de apoio à decisão

Neste ponto analisar-se-ão as funcionalidades de apoio à decisão concebidas e desenvolvidas para a aplicação que se tem vindo a descrever. Estas funcionalidades, que estão acessíveis a partir do lado esquerdo do ecrã, dependem do povoamento do sistema de informação que se traçou pormenorizadamente no ponto anterior. Assim sendo, assume-se a necessidade de carregar no sistema registos de depósitos, clientes, encomendas e veículos antes de se avançar para as funcionalidades relativas à construção de rotas.

##### 4.4.5.2.1. Percurso entre dois pontos

Esta funcionalidade, apesar de estar longe de ser das mais importantes do sistema, oferece ao decisor informações interessantes sobre os percursos entre os pontos geográficos registados no sistema. Este pode, de forma interativa, selecionar dois pontos (depósitos ou clientes) e visualizar no mapa o trajeto entre eles. Além disso, poderá visualizar qual a distância entre os pontos e o tempo de percurso previsto.



Figura 10 – Percurso entre dois pontos

A funcionalidade de Percurso entre dois pontos permite ao decisor, por exemplo, selecionar um depósito e depois, iterativamente, ir selecionando clientes para formar a sua opinião sobre os melhores percursos a realizar.

#### 4.4.5.2.2. Rota Manual

Esta funcionalidade, que permite ao utilizador definir a rota conforme pretende, dota o sistema de uma flexibilidade que se considera imprescindível num problema real, dada a quantidade de variáveis com que um gestor de rotas deve contar, como por exemplo o conhecimento de uma dada área por parte de um condutor ou a necessidade de um atendimento urgente a um dado cliente.

Aqui, o utilizador do sistema deve introduzir o dia para o qual pretende gerar a rota e o veículo a utilizar para a mesma. No ecrã que pede estes *inputs* o utilizador saberá quais os dias que têm encomendas e quantas encomendas existem para cada dia. A Figura 11 ilustra este ecrã de *inputs* iniciais.

Figura 11 – Inputs iniciais da funcionalidade Rota

Após a seleção da data e do veículo surge um ecrã dividido em três partes distintas, cujo exemplo se pode ver na Figura 12. A parcela central é destinada ao mapa onde se poderá visualizar a rota, com os clientes a visitar e o depósito assinalados. No lado direito do ecrã surge o painel de escolha das encomendas a servir. Neste painel surgem apenas as encomendas que existem para o referido dia, com informação sobre o peso de cada. O utilizador pode selecionar

as que deseja e, à medida que o faz, vai sendo informado do peso total das encomendas selecionadas. O sistema, com base na restrição que é a capacidade do veículo escolhido e no peso total das encomendas selecionadas, não permitirá que o utilizador selecione mais encomendas do que as que o veículo consegue transportar. Quando o utilizador selecionar as encomendas que deseja servir e carregar no botão *Traçar*, o sistema enviará o conjunto de pontos a visitar para o *GoogleMaps* e este retornará a melhor rota possível tendo em conta as distâncias a percorrer. Neste ponto de execução, além de surgir no mapa o percurso a realizar e os pontos de paragem, surgem também no painel do lado direito informações sobre a rota, nomeadamente distâncias e tempos associados a cada segmento da rota, a distância total da rota e o tempo total de viagem previsto. Ainda no painel do lado direito, após o desenho de uma rota válida em que o peso das encomendas não ultrapasse a capacidade do veículo, surge o botão *Validar Rota*. Este botão permite, tal como o seu *label* indica, guardar a rota selecionada no sistema de informação. Esta operação guardará em base de dados informações sobre o veículo a utilizar, as encomendas selecionadas e a ordem de visita das mesmas, e a distância total a percorrer.



Figura 12 – Funcionalidade Rota Manual

No fundo da página surgirão informações úteis sobre o veículo selecionado para a rota, nomeadamente o depósito a que está associado, a marca, a matrícula e a capacidade.

#### 4.4.5.2.3. Rotas Automáticas

A funcionalidade que calcula e apresenta as rotas automáticas, ou otimizadas tendo em conta os algoritmos utilizados, merece especial destaque. Esta utiliza um de dois algoritmos à escolha do utilizador para resolver o VRP de dado dia, tendo ainda o mesmo a opção de acionar um algoritmo de pesquisa local. Apresenta-se de seguida a Figura 13, que representa o ecrã que contém os *inputs* necessários para que se construam automaticamente as rotas para dado dia.

Depósitos      Clientes      Encomendas      Frota

### Gerar Rotas Automaticamente

Data: 2012-11-01 (41 enc.)

Algoritmo de resolução: Algoritmo 1

Sem Pesquisa Local

Gerar Rotas

*Algoritmo 1 - Inicia processo de geração de cada rota com base nos clientes com encomendas não visitados. Para cada veículo, é utilizado o algoritmo do vizinho mais próximo para gerar as rotas.*

*Algoritmo 2 - Inspirado do algoritmo Sweep, utiliza agrupamentos (clusters) de clientes para a definição das rotas. Os veículos são escolhidos para as rotas baseados na sua capacidade e nas encomendas existentes no cluster. Após a escolha do veículo é utilizado o algoritmo do vizinho mais próximo para construir a rota.*

*Pesquisa Local - Algoritmo que se aplica após um dos anteriores. Verifica, para cada rota gerada, se existe alguma outra combinação dos pontos a visitar que melhore o resultado (distância a percorrer).*

Home

Universidade do Minho  
Departamento Produção e Operações

Figura 13 – Ecrã de *inputs* para rotas automáticas

Neste ecrã de *inputs* é de salientar que o sistema filtra à partida quais as datas com encomendas e indica o número de encomendas para cada data. Tal evita inconsistências de submissão de datas sem encomendas ou inválidas.

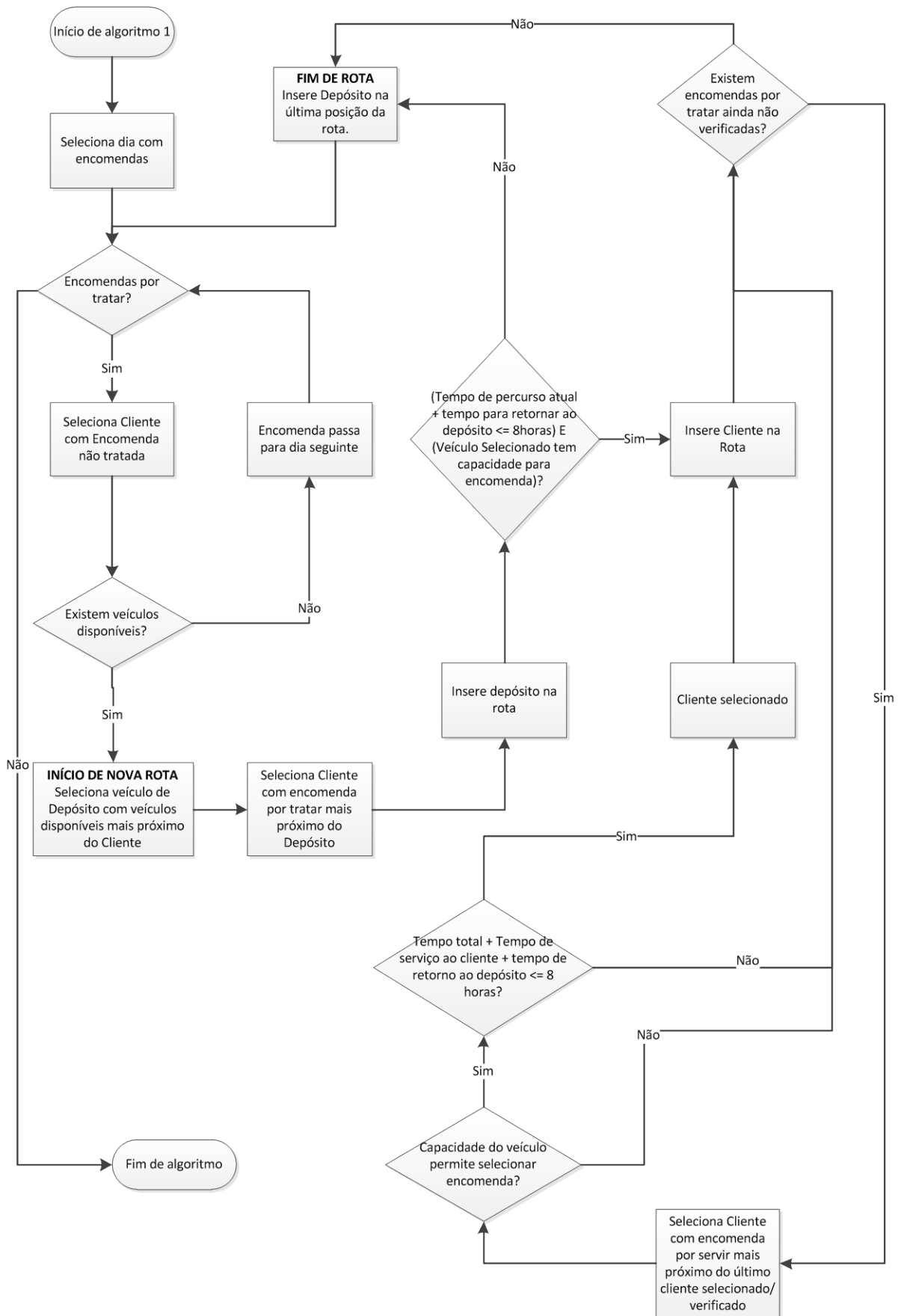
Neste ponto é importante a explicação do funcionamento dos algoritmos passíveis de ser utilizados para a construção das rotas.

#### **4.4.5.2.3.1. Algoritmo 1**

O *Algoritmo 1* usa as encomendas não servidas para iniciar o processo. Seleciona uma encomenda não servida qualquer e escolhe um veículo livre do depósito mais próximo. Depois do veículo selecionado vai sendo aplicada a heurística do vizinho mais próximo para selecionar cada cliente com encomenda a tratar, aplicando-se as restrições da capacidade do veículo e do tempo de viagem, que deverá ser inferior a oito horas estabelecidas. Na fase final do algoritmo, quando uma encomenda não pode ser tratada por não caber no veículo ou por ultrapassar o tempo permitido, o sistema procurará outra encomenda que possa ser incorporada na rota.

Apresenta-se de seguida o fluxograma deste algoritmo. O pseudo-código correspondente poderá ser encontrado no Anexo 1.

## Algoritmo 1 - Fluxograma





A análise dos resultados obtidos pelo *Algoritmo 1* será realizada posteriormente.

#### **4.4.5.2.3.2. Algoritmo 2**

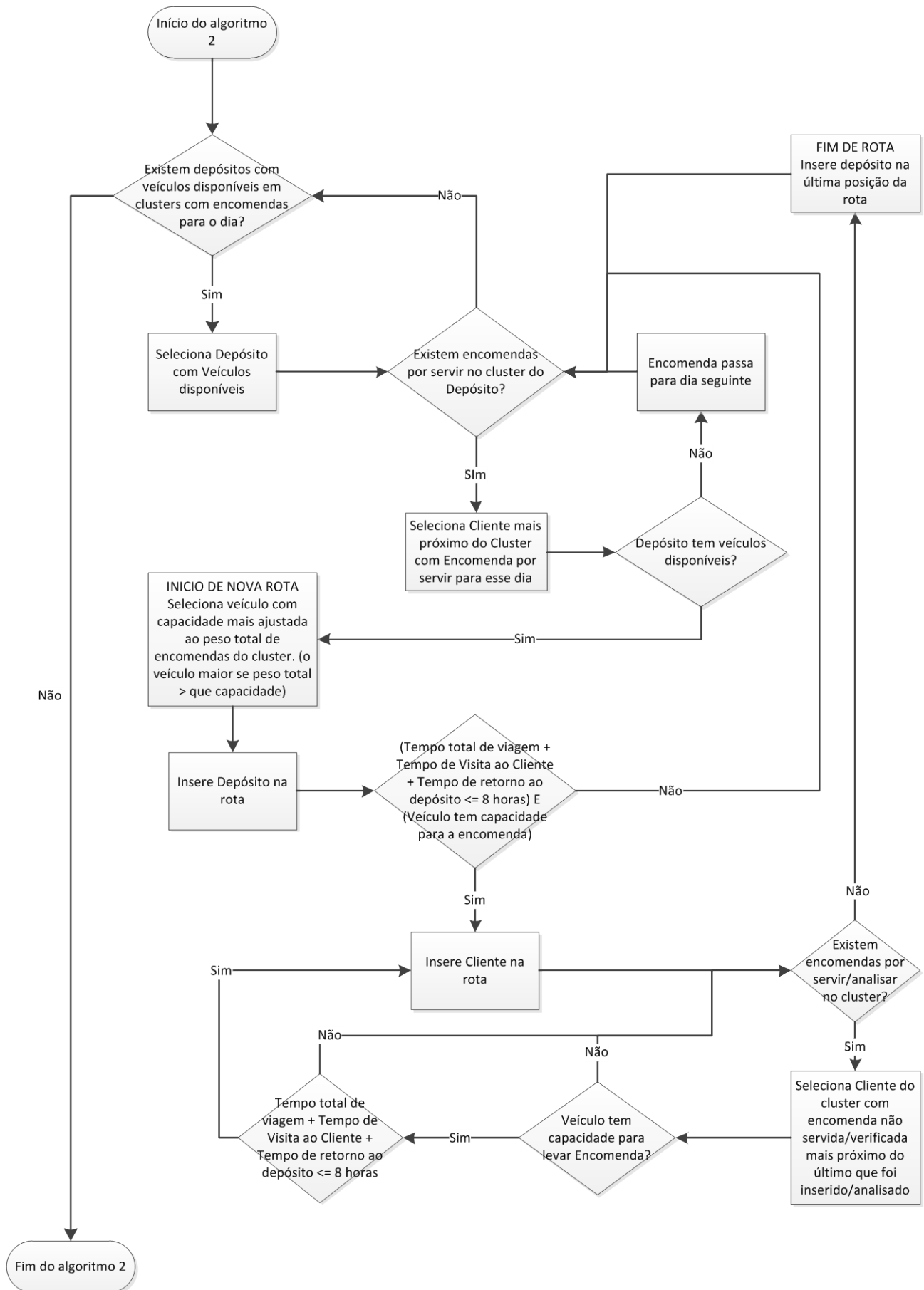
Avançando para a apresentação do *Algoritmo 2*, começa-se por referir que este utiliza o campo *cluster* associado a cada cliente para dividir a área de serviço de cada depósito. Este algoritmo, baseado na anteriormente apresentada heurística *Sweep*, do tipo *Cluster-first, route-second*, foi selecionado exatamente por ser de um tipo diferente do anterior, esperando-se assim obter resultados diferentes. Tal como é característico deste tipo de algoritmos, o mesmo pressupõe que os *clusters* já estão formados, antes de avançar para a construção das rotas. Na realidade, na aplicação desenvolvida, os *clusters* estão sempre formados e atualizados dado que essa operação é realizada aquando da inserção dos pontos geográficos (depósitos ou clientes), conforme se explicou anteriormente.

Este algoritmo, além de ser de um tipo diferente, adiciona ainda uma outra particularidade em que difere do *Algoritmo 1*. Neste caso os veículos não são selecionados ao acaso (pela ordem de inserção na base de dados), como acontece no primeiro algoritmo analisado. Aqui é feita uma análise ao total de encomendas por tratar no *cluster* e é selecionado o veículo com a capacidade mais ajustada. Se este aspeto não fosse tido em conta, o sistema poderia colocar dois veículos mais pequenos a servir os clientes de dado dia, quando existia um maior no depósito que poderia realizar todos os serviços; ou, pelo contrário, poderia selecionar um veículo muito grande para as encomendas existentes, havendo no depósito um veículo mais pequeno onde as mesmas caberiam. Em ambos os casos, esta opção traduzir-se-á numa poupança de veículos utilizados e/ou combustível.

É ainda de salientar que, tal como já acontecia no *Algoritmo 1*, na situação de fecho de cada rota, caso dada encomenda não caiba no veículo ou o tempo de viagem ultrapasse o limite, o sistema procurará nas outras encomendas não servidas do *cluster* alguma que obedeça a estas restrições e possa ser incluída na rota.

Apresenta-se de seguida o fluxograma do *Algoritmo 2*. O pseudo-código poderá ser encontrado no Anexo 2.

## Algoritmo 2 - Fluxograma



A utilização de *clusters* “pura”, apesar de fazer adivinhar as vantagens que existem por as rotas serem mais concentradas nas respetivas áreas, também permite antever desvantagens como:

- Sair um veículo para ir satisfazer uma única encomenda na fronteira do *cluster*, quando houve um veículo do *cluster* vizinho que lá passou perto e poderia ter realizado o serviço.
- Deixar de atender clientes por não ter veículos disponíveis no *cluster*, havendo veículos disponíveis no *cluster* vizinho.

Porém, situações como a segunda, apresentada como desvantagem, podem também tornar-se em vantagens com ganhos significativos. Imaginemos o caso em que no *cluster* de Barcelona, com um depósito na cidade, não existiam veículos disponíveis para atender um último cliente com uma encomenda pequena e que o único *cluster* com veículos disponíveis era o de Braga. Provavelmente adiar a encomenda para o dia seguinte será mais vantajoso para a empresa, do que ir o veículo de Braga satisfazê-la.

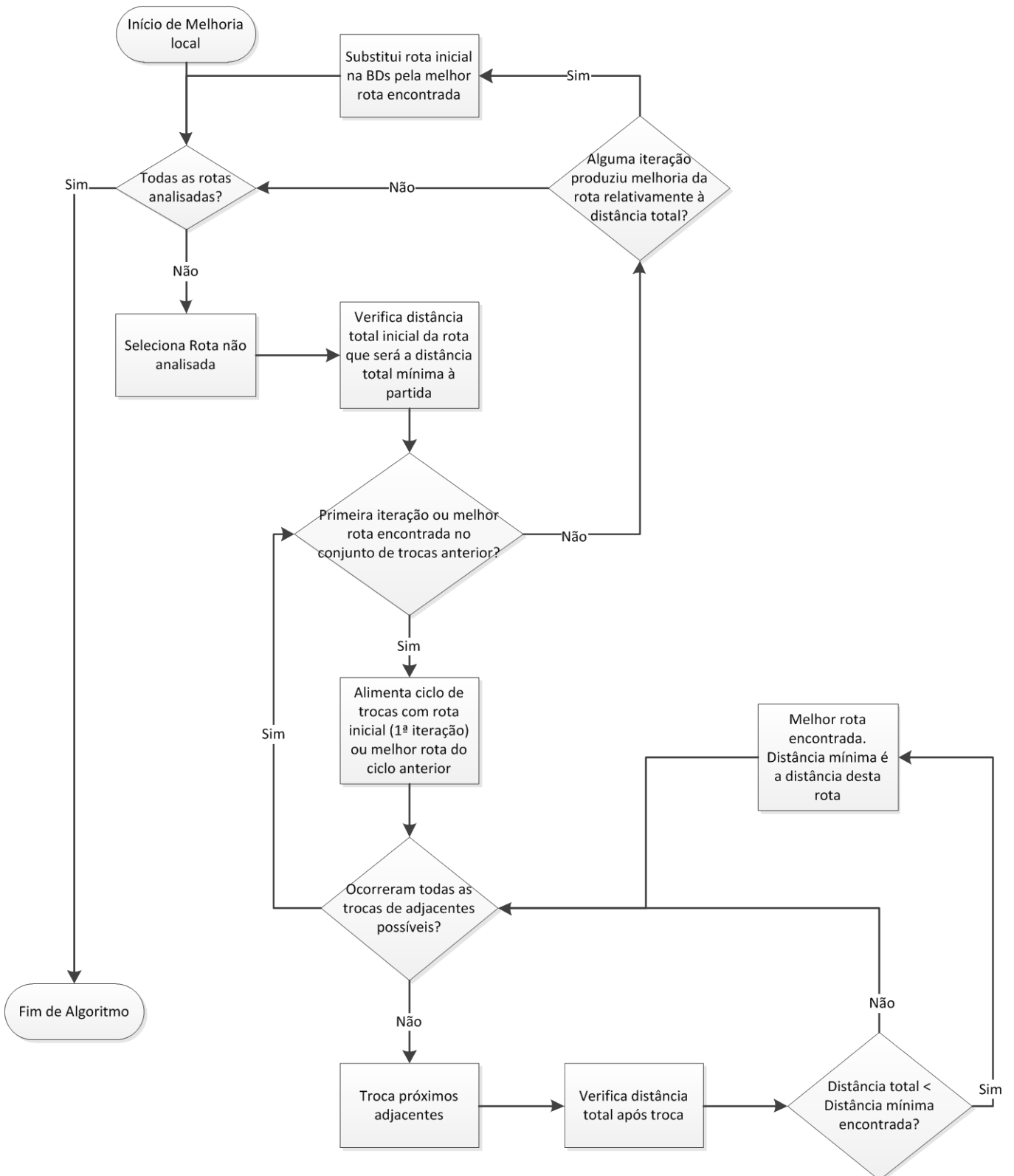
Apresentar-se-ão mais considerações sobre este assunto quando for realizada a análise dos testes e resultados da aplicação.

#### **4.4.5.2.3.3. Algoritmo de Pesquisa Local – *Best Improvement***

Conforme referido anteriormente, além da construção das rotas utilizando os algoritmos apresentados, considerou-se relevante tentar obter os resultados ótimos locais das soluções resultantes. Para tal incorporou-se na aplicação a opção de realizar, após resolver o VRP com um dos algoritmos anteriormente explicados, um algoritmo de pesquisa local – no caso, *Best Improvement*. O procedimento deste algoritmo baseia-se na troca iterativa dos elementos adjacentes de cada rota, procurando um melhor resultado. Para cada uma delas, a solução vinda do algoritmo anterior (*Algoritmo 1* ou *Algoritmo 2*) alimentará o primeiro procedimento de trocas e, a melhor solução gerada (solução que leva a rota a ter menos km) alimentará o procedimento seguinte, ou seja, enquanto existir melhoria o procedimento será alimentado com a "melhor melhoria". Quando não houver melhoria em nenhuma troca, caso a última solução que alimentou o procedimento seja diferente da que está na base de dados, troca-se a solução da base de dados por esta.

Apresenta-se de seguida o fluxograma deste algoritmo. O pseudo-código poderá ser encontrado no Anexo 3.

### Pesquisa Local - Fluxograma



A análise da execução deste algoritmo será realizada aquando da análise de resultados.

Após a apresentação dos algoritmos que gerarão rotas automaticamente e melhorarão os resultados obtidos, avança-se para a exposição de como a aplicação desenvolvida os aplica. Para tal foram criados quatro exemplos de execução que incorporam quatro baterias de testes diferentes:

- Exemplo de execução 1: todos os clientes na base de dados têm encomendas;
- Exemplo de execução 2: apenas os clientes próximos dos depósitos têm encomendas;
- Exemplo de execução 3: apenas os clientes mais afastados dos depósitos têm encomendas;
- Exemplo de execução 4: as encomendas existentes para dado dia ultrapassam a capacidade total dos veículos;

Os dados constantes na base de dados, que serão comuns para todos os testes são:

- 3 depósitos registados: Braga, Coimbra e Lisboa;
- Cada depósito contém os seguintes 3 veículos (total de 9 veículos em Base de Dados) que se caracterizam pelas suas capacidades:
  - Braga: 1 veículo com capacidade de 3000kg e 2 com capacidade de 3500kg;
  - Coimbra: 1 veículo com capacidade de 3500kg e 2 com capacidade de 3000kg;
  - Lisboa: 3 veículos com capacidade de 3500kg
- 41 clientes registados: 21 situam-se relativamente perto dos depósitos e 20 nas extremidades do país.

Nos dados anteriores podemos inferir que a capacidade total de carga é de 30000kg. Tendo em conta estes dados avança-se para a apresentação dos testes realizados e para a análise dos resultados obtidos.

#### **4.4.5.2.3.4. Exemplo de Execução 1**

Neste exemplo de execução, utilizar-se-á um dia em que todos os clientes (41) têm encomendas de 600kg (total de 24600kg). Para começar, no ecrã de *input* (Figura 13),

selecionar-se-á a opção *Algoritmo 1* e *Sem Pesquisa Local*. O resultado é o apresentado na Figura 14.

Sendo a primeira vez que surge a janela que apresenta a resolução do VRP, avança-se para a explicação da mesma. Como se pode observar, no topo da janela surge o algoritmo utilizado e o dia para o qual foram geradas as rotas. A parte central do ecrã será ocupada pelo mapa. Este conterà as rotas desenhadas com cores diferentes para que seja possível identificá-las, assim como aos depósitos de onde saem e os clientes que visitam. Também no mapa encontram-se assinalados diferenciadamente os clientes e os depósitos. Sobre o mapa, é ainda de referir que este tem as funcionalidades de *zoom* ativas para permitir ao utilizador ver pormenorizadamente cada rota.

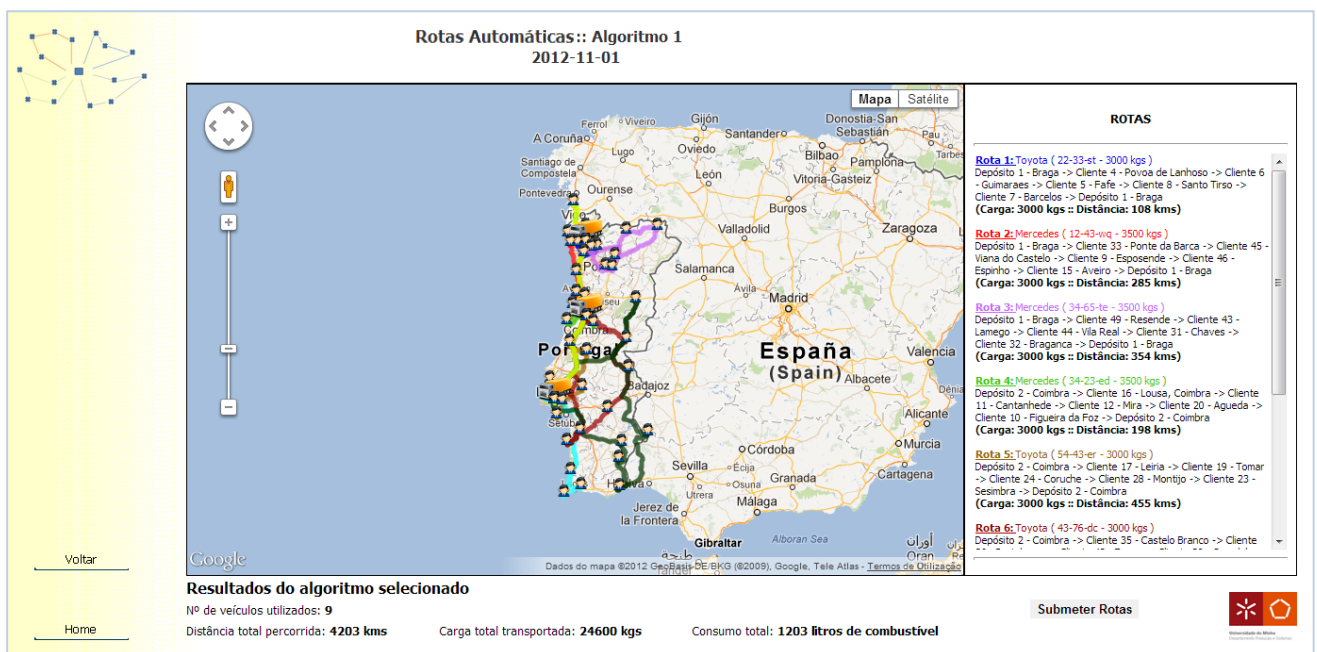


Figura 14 – Algoritmo 1: Todos os clientes com encomendas

No lado direito da janela surge um painel textual com a caracterização de cada rota gerada, ligando cada uma à rota desenhada no mapa através da cor do seu cabeçalho. A informação constante nesse painel é, para cada rota, qual o veículo selecionado, a sua capacidade, todos os pontos visitados na rota (inclusivamente o depósito de onde sai e para onde volta), a carga da rota (soma dos pesos das encomendas) e a distância percorrida. Quanto ao painel do fundo do ecrã, esse é caracterizado por conter os resultados obtidos com a utilização do algoritmo. Refere o número de veículos utilizados, a distância total percorrida no conjunto de todas as rotas, a carga total transportada e faz uma estimativa do consumo total de combustível, baseado na distância total percorrida em cada rota e no consumo médio de cada veículo (valor inserido no

sistema de informação aquando do registo de cada veículo). Para concluir a apresentação do painel de resultados, resta mencionar o botão *Submeter Rotas*. Este botão realizará a validação das rotas apresentadas, submetendo-as na base de dados (ou, especificando com mais exatidão, copiando-as das tabelas temporárias *Rotas\_Temp* e *Enc\_Rotas\_Temp* para as tabelas finais *Rotas* e *Enc\_Rotas*).

Após a apresentação do ecrã de resultados, avança-se para a análise do resultado desta primeira execução, lembrando que foi utilizado o *Algoritmo 1* sem pesquisa local. Olhando para a mancha de rotas, facilmente se averigua que as mesmas se cruzam e misturam, havendo mesmo um caso flagrante de uma rota amarelo-fluorescente que se consegue reparar que vai de Lisboa a Vigo (havendo um depósito em Braga). No painel das rotas, essa não aparece na imagem pois, na realidade, foi a última a ser construída (aparece quando rodamos o *scrool* do painel para baixo). Tendo sido a última a ser construída, deu-se o caso de, para o cliente que faltava servir – Vigo – o único veículo que restava disponível no sistema era de Lisboa. Começa-se a verificar uma característica de utilizar meramente a heurística do vizinho mais próximo. A palavra *caraterística* foi utilizada ao invés de *problema*, por o facto apontado, em casos particulares poder não ser um entrave.

Apresenta-se de seguida o resultado da execução do *Algoritmo 2* para o mesmo dia (mesmas encomendas) e também sem pesquisa local, para que se possam analisar as diferenças.

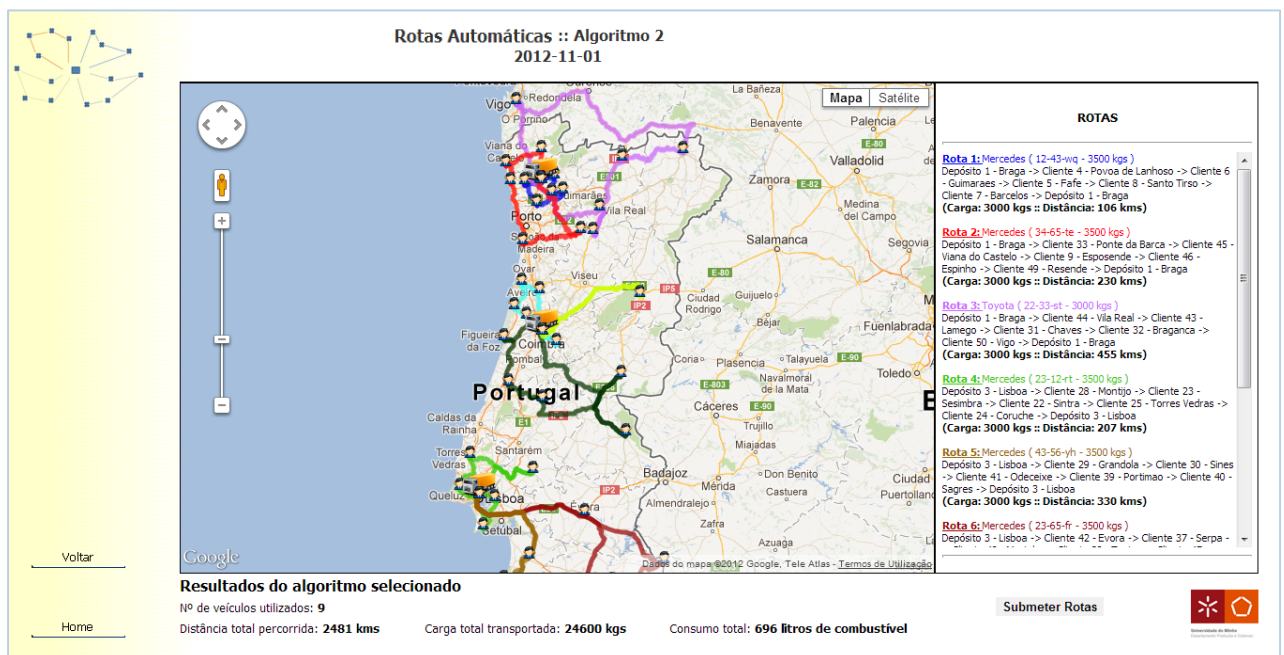


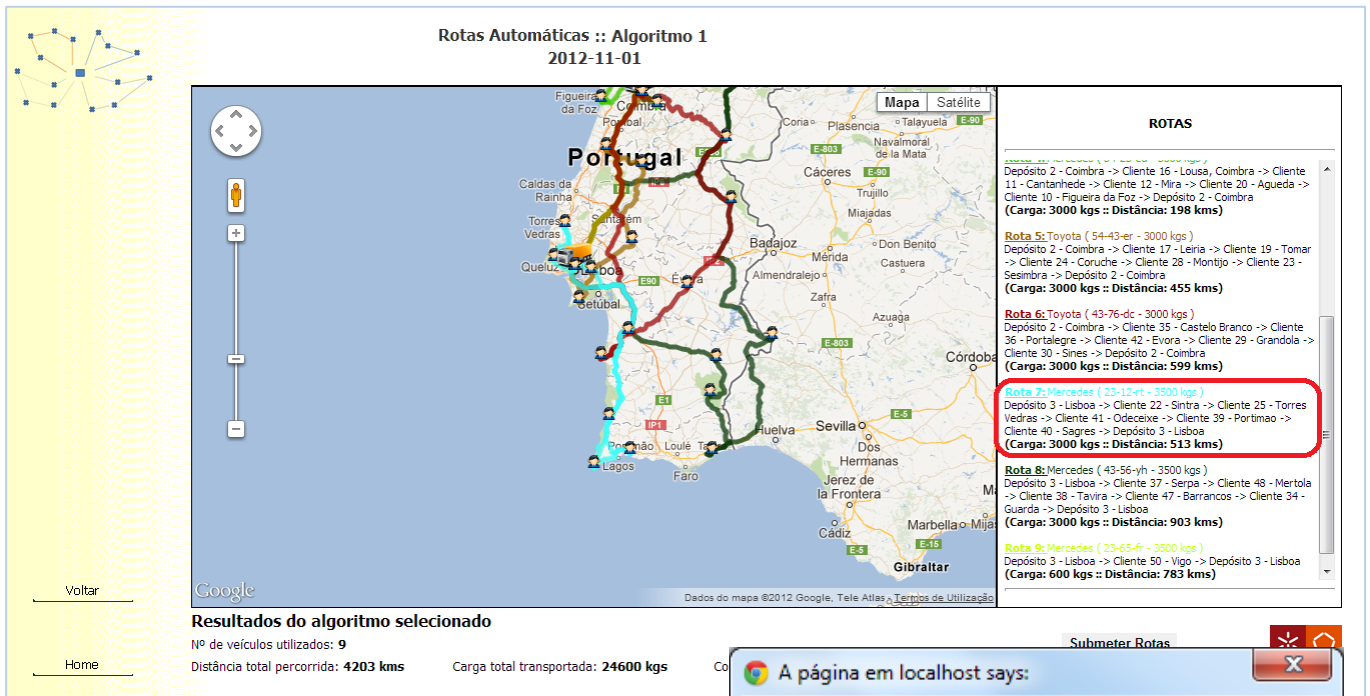
Figura 15 – Algoritmo 2: Todos os clientes com encomendas

A utilização do *Algoritmo 2* oferece, logo à primeira vista, rotas mais “arrumadas”. Não se vêem cruzamentos de rotas vindas de *clusters* diferentes, apresentando-se manchas de rotas mais organizadas, e a primeira ideia é que este algoritmo cria um melhor conjunto de rotas. Analisando os resultados no painel existente na base do ecrã e comparando-os com os do *Algoritmo 1* chega-se à conclusão que, para a mesma carga a transportar, o *Algoritmo 2* apresenta resultados consideravelmente melhores, satisfazendo as mesmas encomendas e realizando menos 1722km. Como é natural, dada a forma como é calculada, a estimativa de consumo de combustível também é consideravelmente menor.

Além deste aspeto, pode-se reparar na alocação de veículos. Segundo o descrito aquando da explicação do algoritmo, deveria ser selecionado o veículo mais ajustado ao conjunto de encomendas. Tal parece não se verificar porém, como o total das encomendas ultrapassa as capacidades individuais dos veículos, o algoritmo vai selecionando os veículos com mais capacidade primeiro, numa tentativa de minimizar o número de veículos utilizados. Caso o peso total das encomendas fosse inferior à capacidade individual dos veículos, aí sim, procurar-se-ia o menor veículo capaz de as transportar, ajustando o veículo às encomendas.

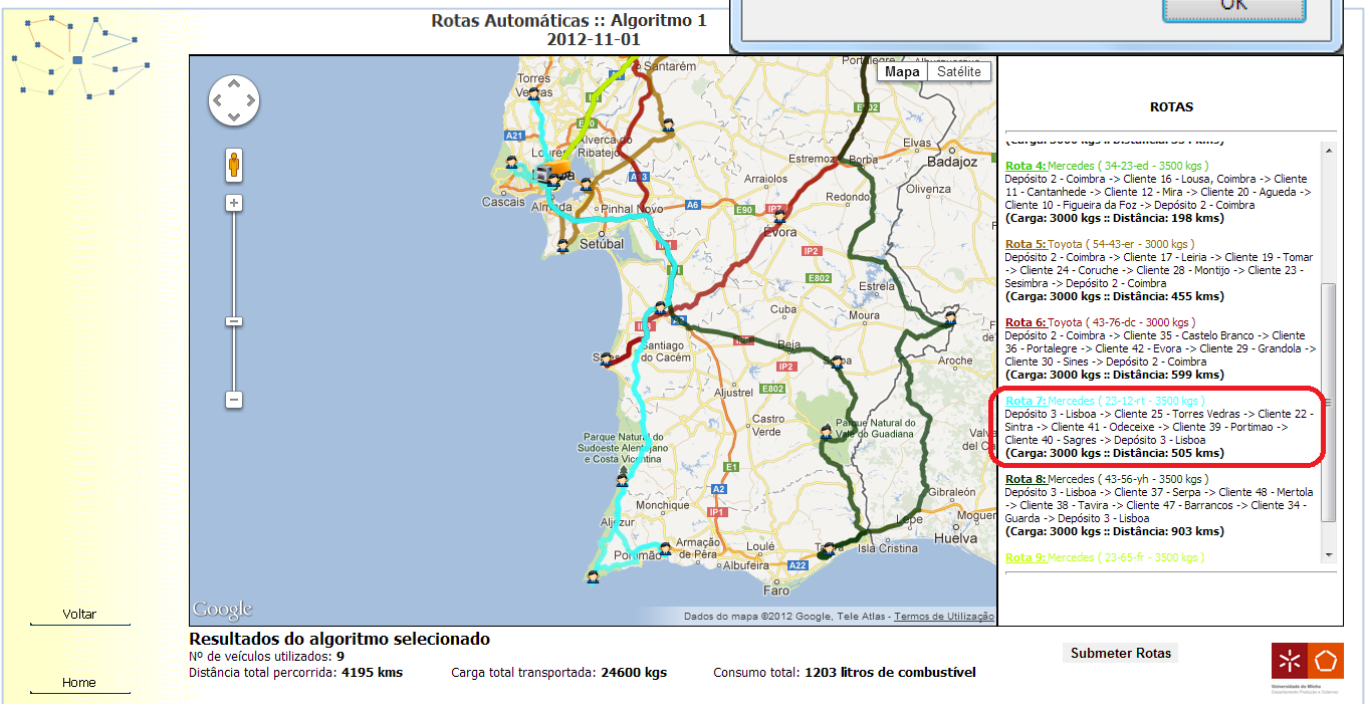
Resta a aplicação do algoritmo de pesquisa local. Porém, como é normal, este nem sempre melhora a solução e é isso que acontece com a aplicação deste algoritmo ao resultado do *Algoritmo 2*. No entanto, a pesquisa local em causa melhora a solução do *Algoritmo 1* e a Figura 16 traduz isso mesmo. Como se vê na referida figura, quando o algoritmo de pesquisa local encontra uma solução melhor do que a encontrada pelo algoritmo que o alimentou (*Algoritmo 1*), o sistema informa o utilizador do local da melhoria (rota em que a mesma acontece) e de quanto é essa melhoria. Na referida figura apresenta-se a aplicação do *Algoritmo 1* sem pesquisa local e com pesquisa local, para ser possível a análise da diferença.





A página em localhost says:  
**Best Improvement: Rota 7 melhorada em 7.95kms.**

OK

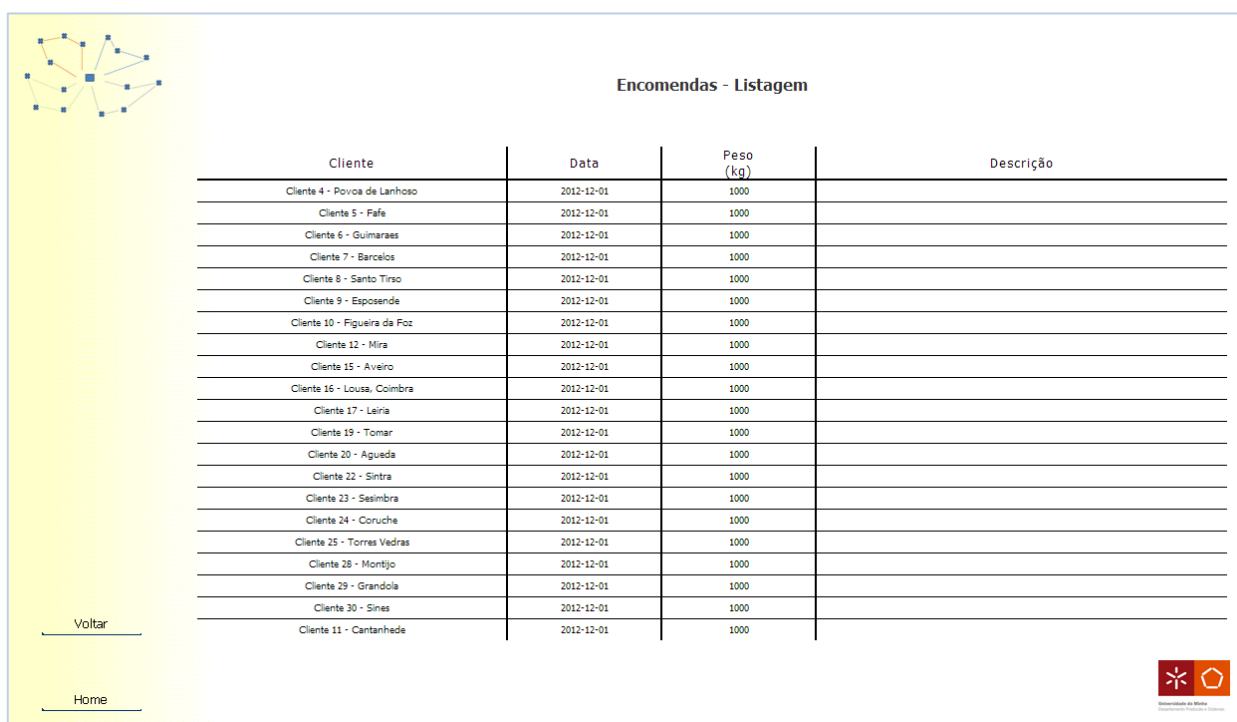


**Figura 16 – Pesquisa local aplicada ao Algoritmo 1: Todos os clientes com encomendas**

Após a explicação pormenorizada do ambiente de apresentação dos resultados dos algoritmos associada à análise do primeiro exemplo de execução, avança-se para a exposição e análise dos resultados dos testes seguintes.

#### 4.4.5.2.3.5. Exemplo de Execução 2

O segundo exemplo de execução tem como *input* para os algoritmos um dia com menos encomendas, utilizando os clientes que se situam mais perto dos depósitos. Começa por se apresentar o conjunto das referidas encomendas, utilizando a funcionalidade de listagem de encomendas da aplicação (no exemplo de execução anterior optou-se por se caracterizar o *input*, ao invés de o apresentar).



Cliente	Data	Peso (kg)	Descrição
Cliente 4 - Povoia de Lanhoso	2012-12-01	1000	
Cliente 5 - Fafe	2012-12-01	1000	
Cliente 6 - Guimarães	2012-12-01	1000	
Cliente 7 - Barcelos	2012-12-01	1000	
Cliente 8 - Santo Tirso	2012-12-01	1000	
Cliente 9 - Espoende	2012-12-01	1000	
Cliente 10 - Figueira da Foz	2012-12-01	1000	
Cliente 12 - Mira	2012-12-01	1000	
Cliente 15 - Aveiro	2012-12-01	1000	
Cliente 16 - Lousa, Coimbra	2012-12-01	1000	
Cliente 17 - Leiria	2012-12-01	1000	
Cliente 19 - Tomar	2012-12-01	1000	
Cliente 20 - Agueda	2012-12-01	1000	
Cliente 22 - Sintra	2012-12-01	1000	
Cliente 23 - Sesimbra	2012-12-01	1000	
Cliente 24 - Coruche	2012-12-01	1000	
Cliente 25 - Torres Vedras	2012-12-01	1000	
Cliente 28 - Montijo	2012-12-01	1000	
Cliente 29 - Grandola	2012-12-01	1000	
Cliente 30 - Sines	2012-12-01	1000	
Cliente 11 - Cantanhede	2012-12-01	1000	

Figura 17 – Listagem de encomendas que servirão de *input* para os algoritmos no segundo exemplo de execução

Para estas encomendas, a utilização do *Algoritmo 1*, sem pesquisa local, dá o resultado que se segue.

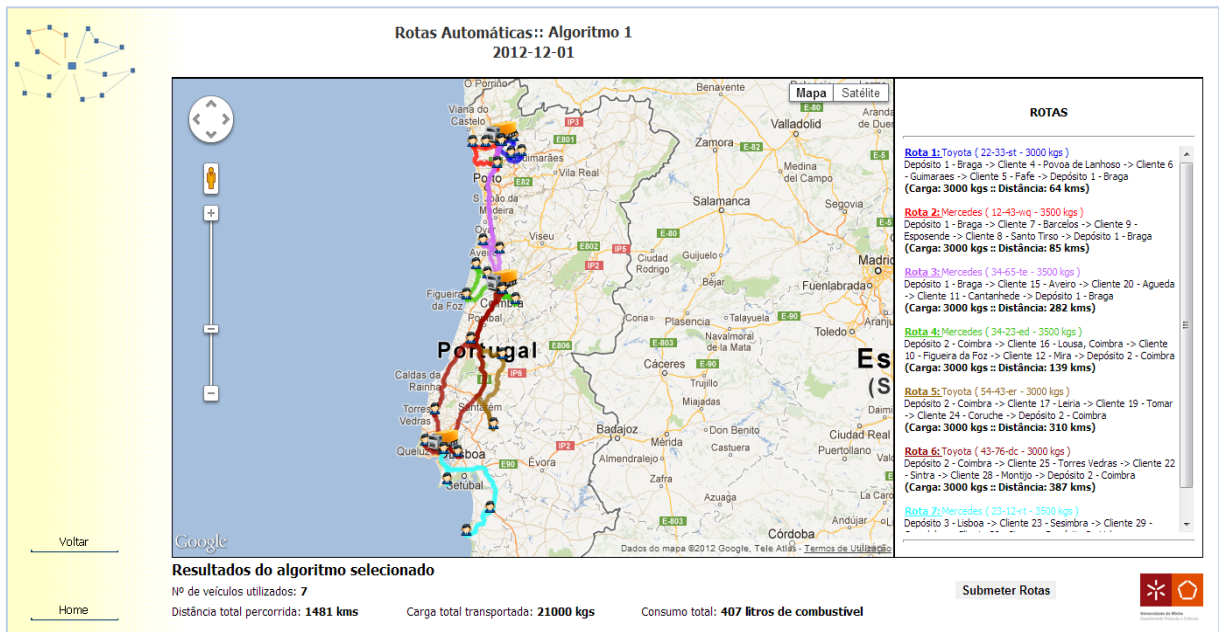


Figura 18 – Algoritmo 1 aplicado ao segundo input de testes

Este exemplo de execução, traduzido pela Figura 18, permite analisar mais pormenorizadamente a aplicação do *Algoritmo 1*. Repare-se nas rotas que saem do depósito de Braga, que se verifica ser o primeiro depósito para o qual o sistema as gera, pela ordem em que surgem no painel de rotas. As duas primeiras parecem boas soluções, mas na terceira, como o veículo tem uma capacidade superior ao peso das encomendas que restam na zona, vem para a região de Coimbra procurar as encomendas que completem a sua capacidade. Daí o facto de, no uso deste algoritmo, as rotas de cruzarem. Apresenta-se de seguida a execução do *Algoritmo 2*, sem pesquisa local, para o *input* em causa.

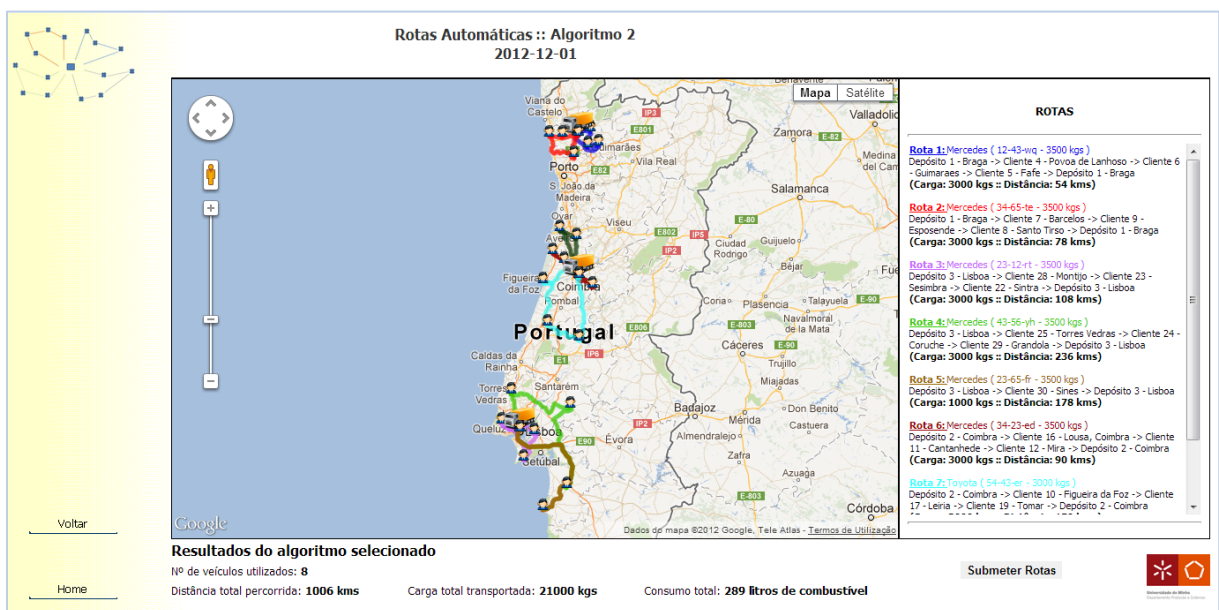


Figura 19 – Algoritmo 2 aplicado ao segundo input de testes


A análise do resultado do *Algoritmo 2*, traduzido na Figura 19, mostra que, além da melhoria em termos distância total percorrida e da previsão de consumo de combustível (que já se tinham verificado no *Exemplo de Execução 1*), este é conceptualmente diferente do *Algoritmo 1*. O facto de os clientes estarem agrupados em *clusters* impede que os algoritmos tenham comportamentos que são muitas vezes indesejados, como o que acontece no caso do exemplo de execução do *Algoritmo 1*, em que o veículo para preencher a sua capacidade vai atender um cliente que se encontra perto de outro depósito.

Porém é de salientar que nem todos os parâmetros do *Algoritmo 2* apresentam melhorias relativamente ao *Algoritmo 1*. Repare-se que o número de veículos utilizados para a realização das rotas é inferior no caso do *Algoritmo 1*, e esse pode ser um aspeto de grande relevo, dados os custos de manutenção de veículos na frota.

No caso das encomendas submetidas para estes testes a pesquisa local não melhorou a solução de nenhum algoritmo e, por essa razão, não existem resultados diferentes dos anteriores para apresentar.

#### **4.4.5.2.3.6. Exemplo de Execução 3**

O *input* deste exemplo difere do anterior no posicionamento dos clientes a servir e no peso total das encomendas. No anterior os clientes situavam-se junto aos depósitos e neste encontram-se posicionados nos extremos do país e, quanto ao peso total das encomendas, no exemplo anterior era 21000kg (21 encomendas de 1000kg) e agora é 12000kg (20 encomendas de 600kg). Apresenta-se de seguida o conjunto das encomendas que formarão o *input* para os algoritmos neste exemplo de execução.



### Encomendas - Listagem

Cliente	Data	Peso (kg)	Descrição
Cliente 31 - Chaves	2013-01-01	600	
Cliente 32 - Braganca	2013-01-01	600	
Cliente 33 - Ponte da Barca	2013-01-01	600	
Cliente 34 - Guarda	2013-01-01	600	
Cliente 35 - Castelo Branco	2013-01-01	600	
Cliente 36 - Portalegre	2013-01-01	600	
Cliente 37 - Serpa	2013-01-01	600	
Cliente 38 - Tavira	2013-01-01	600	
Cliente 39 - Portimao	2013-01-01	600	
Cliente 40 - Sagres	2013-01-01	600	
Cliente 41 - Odeceixe	2013-01-01	600	
Cliente 42 - Evora	2013-01-01	600	
Cliente 43 - Lamego	2013-01-01	600	
Cliente 44 - Vila Real	2013-01-01	600	
Cliente 45 - Viana do Castelo	2013-01-01	600	
Cliente 46 - Espinho	2013-01-01	600	
Cliente 47 - Barrancos	2013-01-01	600	
Cliente 48 - Mertola	2013-01-01	600	
Cliente 49 - Resende	2013-01-01	600	
Cliente 50 - Vigo	2013-01-01	600	

Voltar

Home




Figura 20 – Listagem de encomendas que servirão de *input* para os algoritmos no terceiro exemplo de execução

Apresentam-se na Figura 21 os resultados da aplicação deste *input* ao Algoritmo 1, sem pesquisa local.



Figura 21 – Algoritmo 1 aplicado ao terceiro *input* de testes

A primeira alteração que se salienta no uso deste algoritmo para o *input* em questão, é que o mesmo apenas utiliza 4 veículos de 2 depósitos para realizar o serviço. No caso de este tipo de



padrões de encomendas suceder com alguma frequência, provavelmente o gestor da empresa deverá começar a pesar os encargos com o terceiro depósito face às mais-valias que o mesmo oferece. Apresenta-se de seguida a aplicação do mesmo *input* ao *Algoritmo 2*, sem pesquisa local.

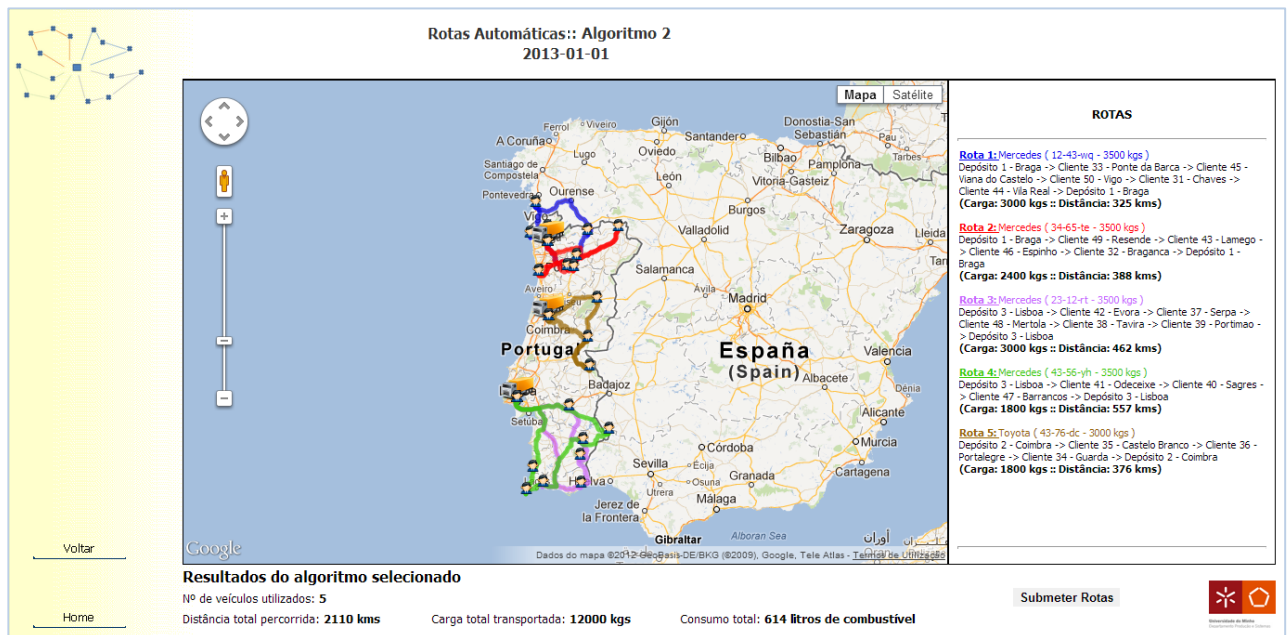


Figura 22 – Algoritmo 2 aplicado ao terceiro *input* de testes

A análise dos resultados da aplicação do *Algoritmo 2* ao terceiro *input* de testes comprova o que já anteriormente se mencionou, de que este tipo de algoritmos que agrupa primeiro os clientes antes de criar as rotas melhora os resultados em termos de distância percorrida e de estimativa de consumo de combustível, relativamente a algoritmos mais simples como a heurística do vizinho mais próximo. Porém, com base no apresentado nos resultados deste exemplo de execução, caberá ao gestor de frota pesar estas vantagens face às desvantagens da utilização de mais veículos (mais uma vez a solução proposta pelo *Algoritmo 2* utiliza mais um veículo que a do *Algoritmo 1*) e de mais um depósito, que implicará com certeza custos como manutenção e aluguer, entre outros.

Para o caso deste *input*, a pesquisa local melhora as soluções dos dois algoritmos. Nitidamente, e como seria de esperar, o fator influenciador da ação deste algoritmo é o posicionamento geográfico dos clientes relativamente uns aos outros, e não pesos das encomendas ou o número de pontos a visitar. Este facto era espectável porque o algoritmo

permuta pontos adjacentes das rotas na procura de melhores resultados em termos de distâncias totais percorridas e, assim sendo, consoante as posições geográficas dos seus clientes, cada rota é passível de ser melhorada. Apresentam-se de seguida os resultados da aplicação da heurística de pesquisa local a cada algoritmo.

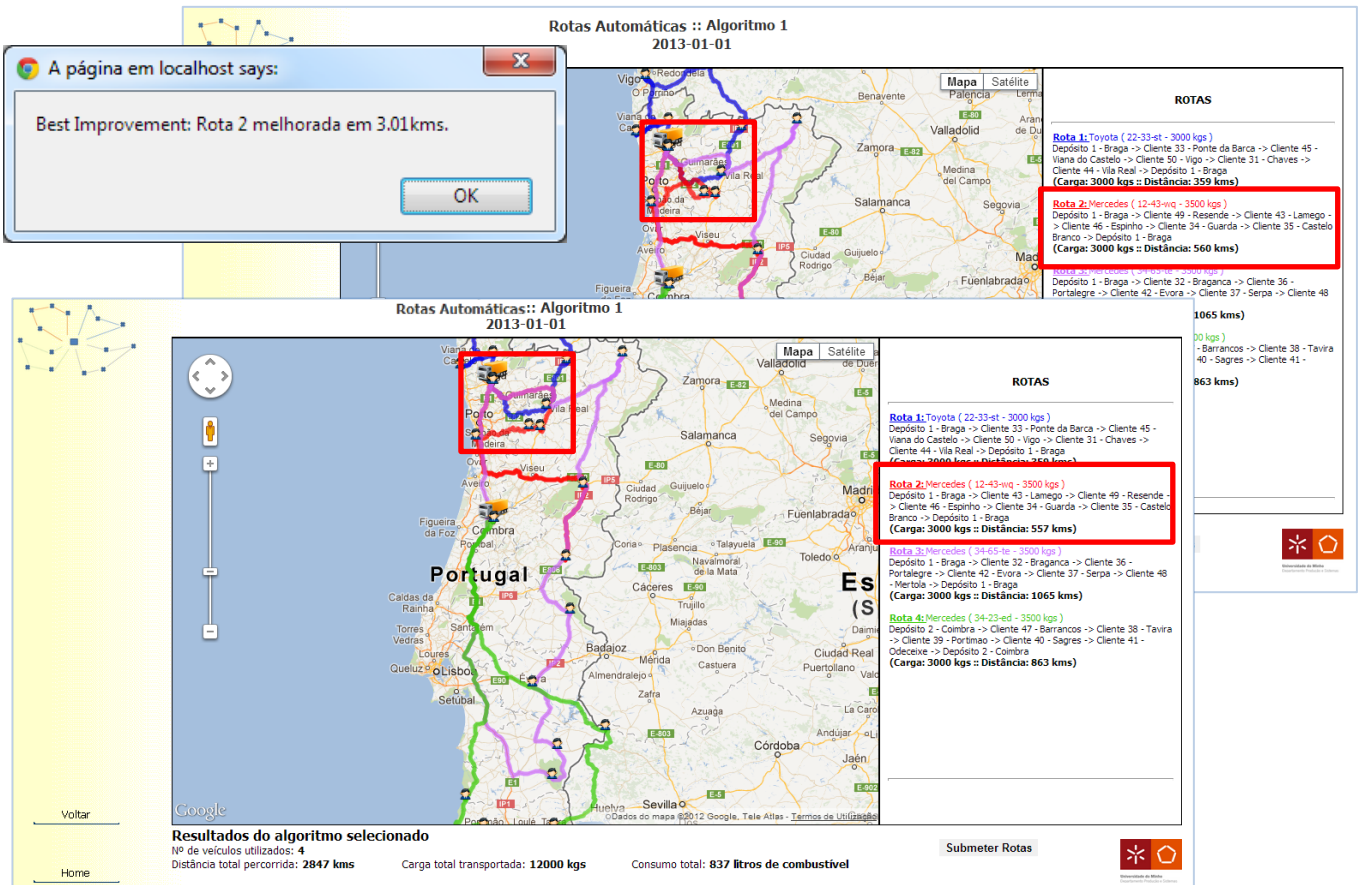


Figura 23 – Best Improvement aplicada ao resultado do terceiro input de testes do Algoritmo 1

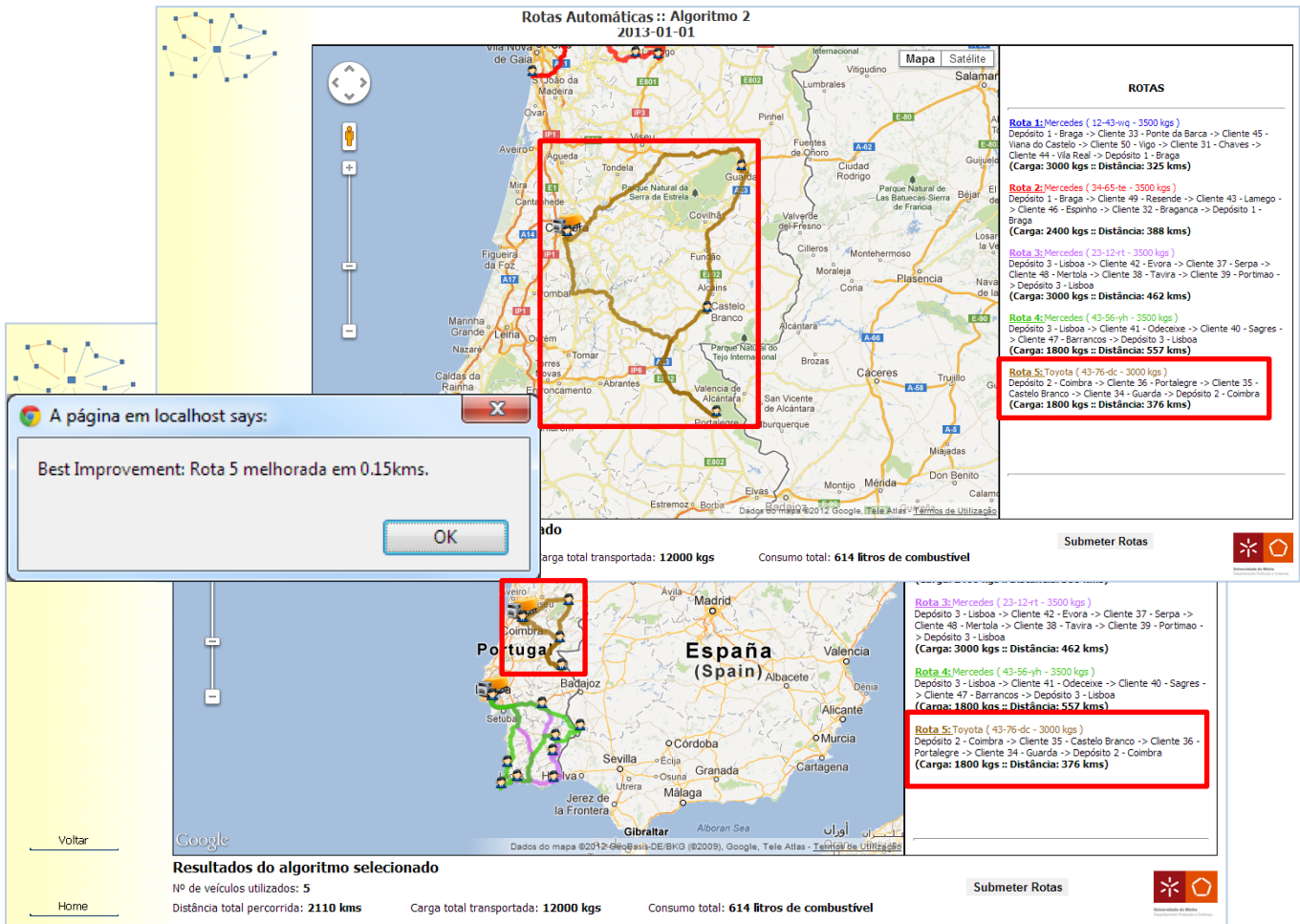


Figura 24 – Best Improvement aplicada ao resultado do terceiro input de testes do Algoritmo 2

#### 4.4.5.2.3.7. Exemplo de Execução 4

Neste último exemplo de execução tentar-se-á analisar um caso ainda não abordado: o comportamento do sistema se o peso total das encomendas for superior à soma das capacidades dos veículos (30000kg). Para tal neste teste alterar-se-ão todas as encomendas do *input* anterior para o peso de 1600kg, o que dará um total encomendado de 32000kg (20 encomendas de 1600kg).

Começar-se-á por analisar o comportamento do *Algoritmo 1*, sem pesquisa local, face a este *input* de encomendas.



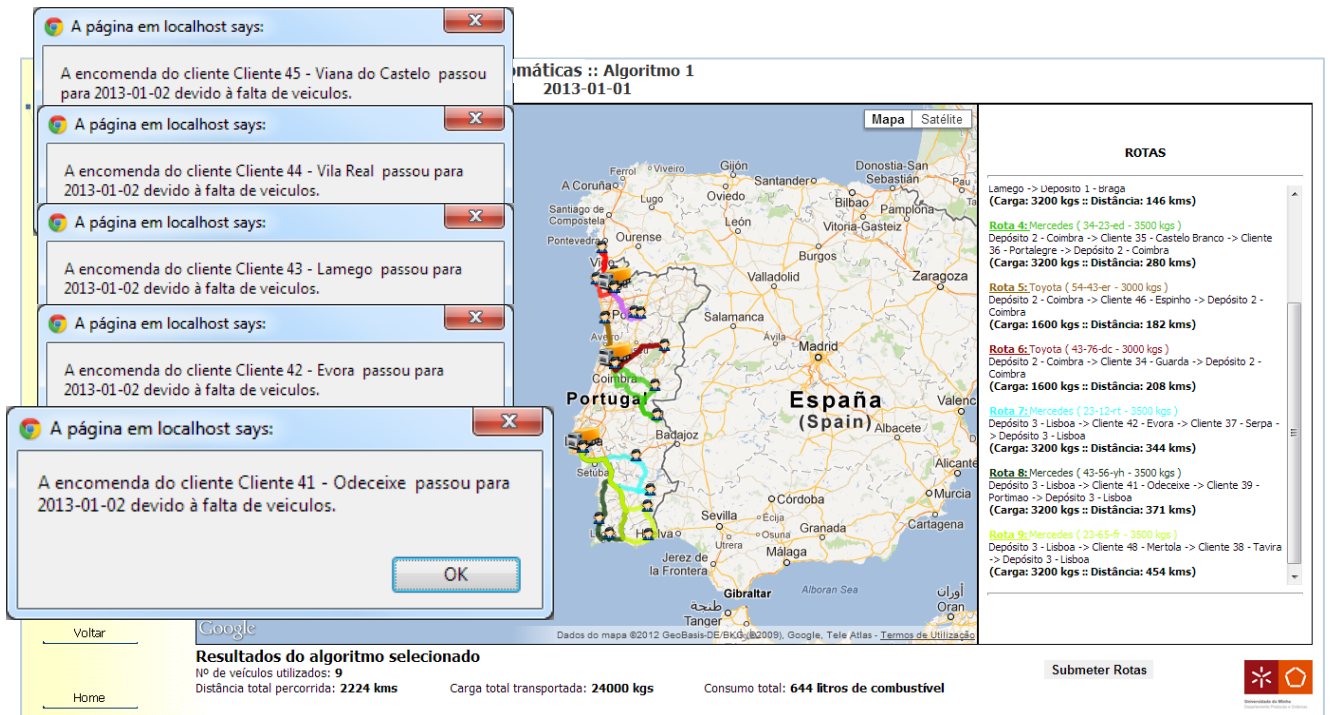


Figura 25 – Algoritmo 1 aplicado ao quarto input de testes

Como se pode ver na Figura 25, o utilizador será informado de todas as encomendas que, por não poderem ser servidas, transitarão para o dia seguinte (procedimento previsto nos fluxogramas e pseudo-código dos algoritmos).

Conforme imposto pelo *input* submetido, nem todas as encomendas puderam ser servidas, estando nesta situação 5 encomendas de 1600kg (8000kg). Este facto levanta a questão: Se a capacidade total dos veículos é de 30000kg e o total de encomendas era de 32000kg, porque razão houve 8000kg de encomendas não servidas? A explicação é simples. O facto de haver alguns veículos com capacidade de 3000kg leva a que cada veículo destes só consiga levar uma encomenda de 1600kg, “desperdiçando” assim 1400kg de capacidade. Este “desperdício” gera o interesse de implementação de um algoritmo onde seja possível ao decisor dividir as encomendas, podendo os clientes ser servidos por mais do que um veículo (maximizando a carga transportada por veículo, logo reduzindo o total de veículos necessários). Porém esta solução também obrigará a pesar desvantagens, de onde se salienta o aumento de consumo de combustível associado a visitar clientes mais do que uma vez para a entrega de uma encomenda repartida.

Apresenta-se de seguida o resultado do *Algoritmo 2* para este *input*, para que seja possível realizar uma análise comparativa dos resultados.

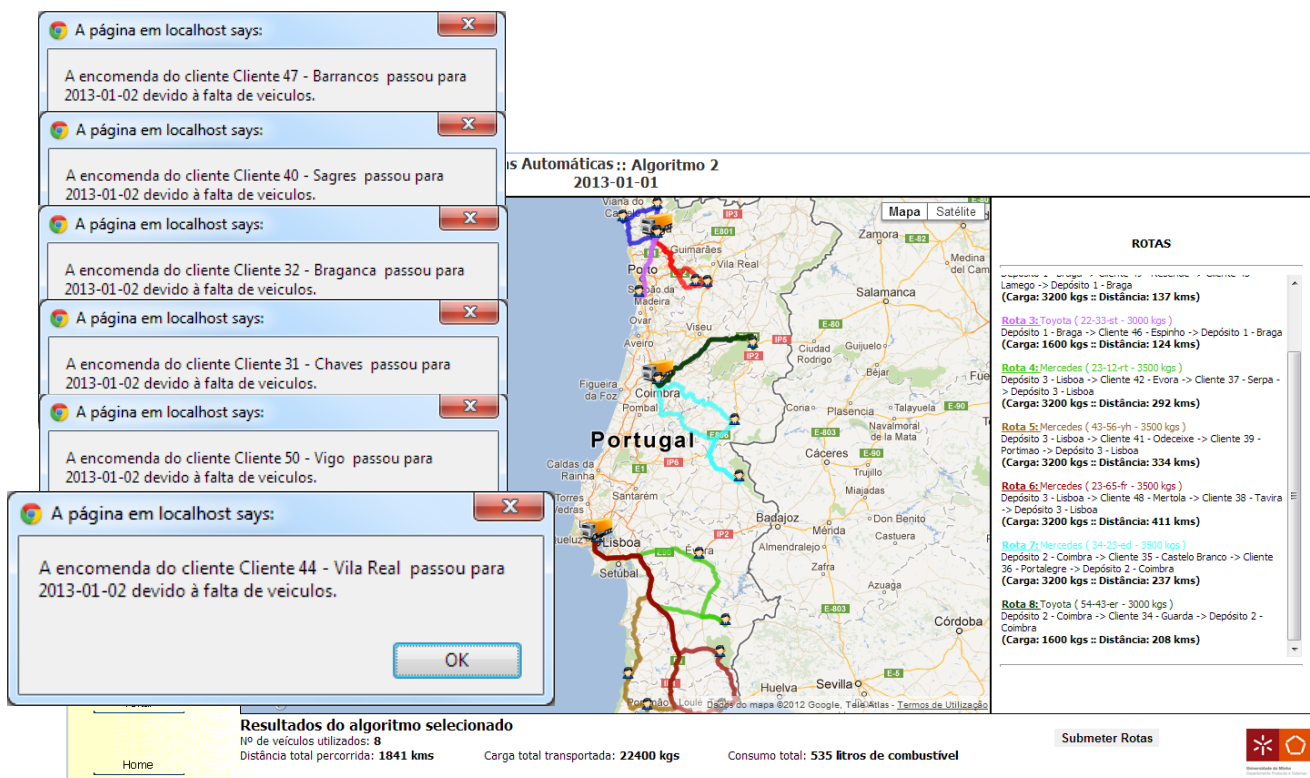


Figura 26 – *Algoritmo 2* aplicado ao quarto *input* de testes

A análise da Figura 26 permite várias observações. A primeira que se salienta é a de que houve mais uma encomenda não servida (6) do que com o *Algoritmo 1* e, um ponto que aliado a este facto é à primeira vista estranho, é de que houve um veículo que não saiu do depósito (apenas foram utilizados 8). A explicação para esta situação é simples. No *Algoritmo 2* os clientes são servidos por veículos do depósito que corresponde ao seu *cluster*. Neste caso, como facilmente se pode verificar pelos clientes não atendidos, o depósito de Coimbra apenas tinha 3 clientes para servir, e nenhum dos que ficaram por servir pertencia ao esse *cluster*. Assim sendo, nenhuma encomenda foi alocada ao terceiro veículo desse depósito, uma vez que os dois anteriores já tinham todas as encomendas do *cluster* colocadas nas suas rotas. Também merece destaque o facto de neste caso a diferença de distâncias percorridas *versus* carga transportada entre os dois algoritmos ser inferior, como a mancha de rotas do *Algoritmo 1* visível na Figura 25 deixava antever.

Apesar de o posicionamento geográfico dos clientes com encomendas ser o mesmo do exemplo anterior, a pesquisa local não ofereceu melhorias aos resultados de nenhum dos algoritmos. Apesar da similaridade aparente em termos de posicionamento geográfico dos

pontos a visitar, o facto é que nem todos são servidos e as rotas construídas que alimentam o algoritmo de pesquisa local são completamente diferentes, explicando-se assim o facto de não haver melhorias.

#### 4.4.5.2.4. Visualizar/Apagar Rotas

Após a análise dos resultados dos algoritmos de otimização de rotas, na reta final da apresentação das funcionalidades oferecidas pelo sistema desenvolvido, surge a de *Visualizar/Apagar Rotas*. Esta opção pressupõe que o utilizador já tenha submetido/vinculado rotas geradas manualmente ou automaticamente. Apresentam-se de seguida os ecrãs relacionados com a visualização e eliminação de rotas.

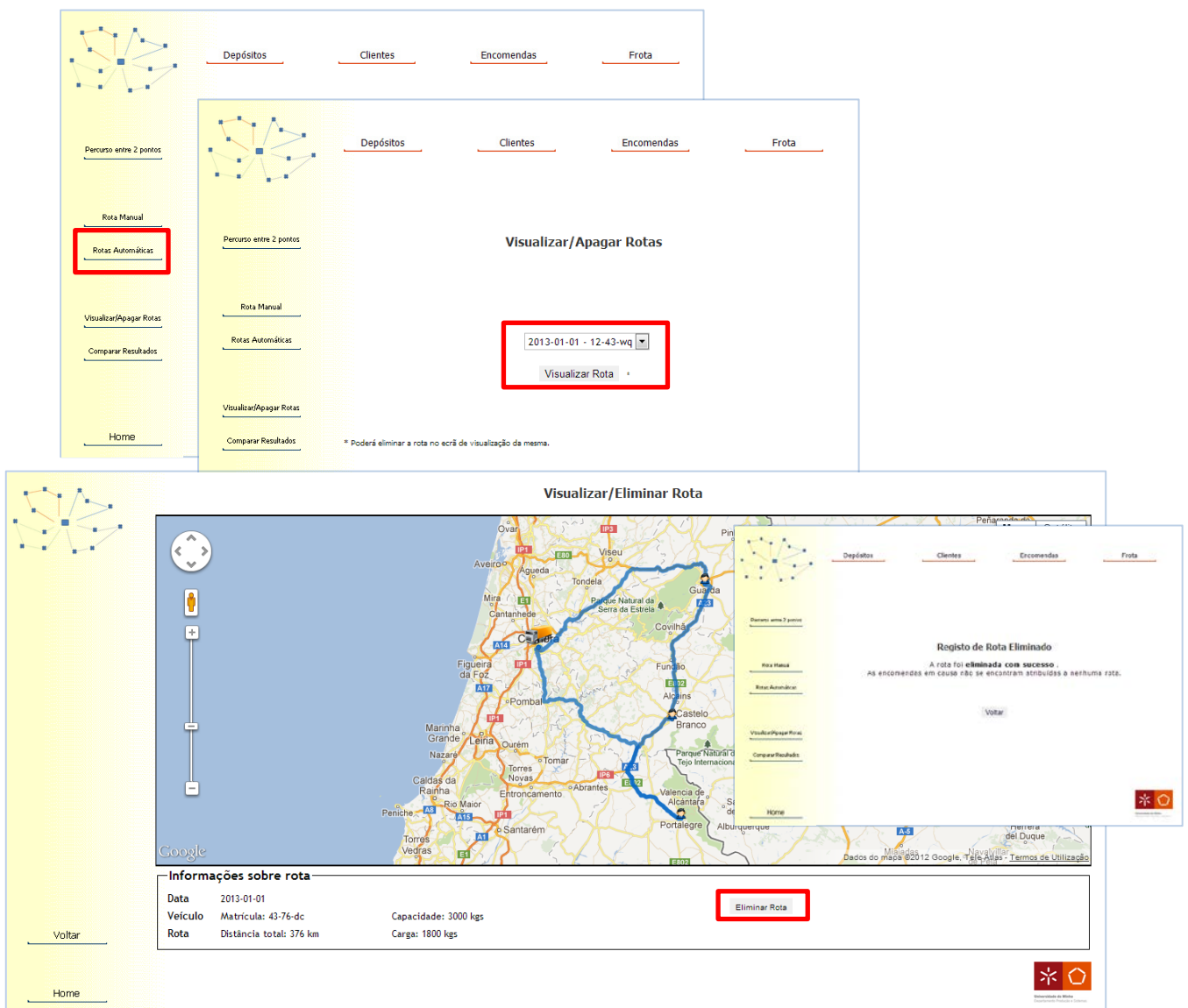


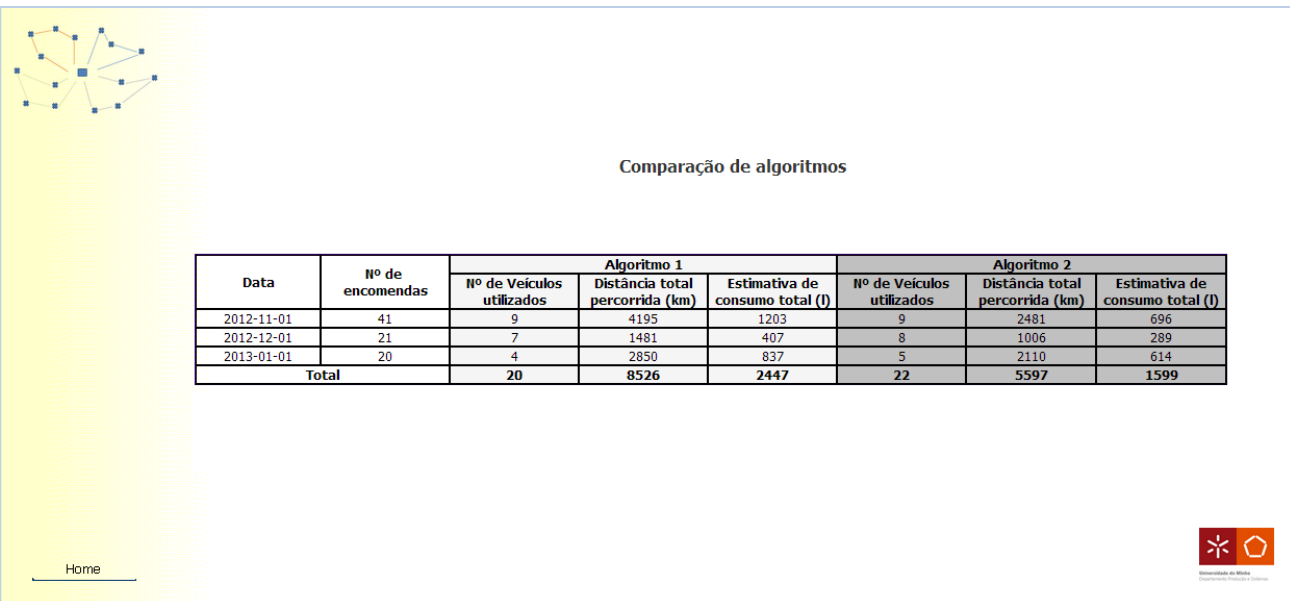
Figura 27 – Conjunto de ecrãs descritivos da funcionalidade de Visualizar/Apagar Rota

Nesta funcionalidade o utilizador seleciona a rota a visualizar/apagar mediante a data e o veículo associados à rota. O ecrã que surge após essa seleção divide-se em duas partes. Na maior parte do ecrã surge um mapa com a rota desenhada e com o depósito e clientes assinalados. Abaixo do mapa surge um painel de informações relativas à rota, nomeadamente a data, a matrícula e capacidade do veículo, a distância percorrida e a carga total transportada. Ainda neste painel de informações aparece um botão que permite eliminar a rota que se está a visualizar.

Esta funcionalidade foi pensada como uma forma de completar a flexibilidade do sistema. O utilizador pode, por exemplo, gerar automaticamente todas as rotas de um dia utilizando um dos algoritmos anteriormente apresentados, mesmo que não tenha intenção de implementar algumas rotas do conjunto total. Nesse caso, poderá utilizar esta funcionalidade para eliminar as rotas que não considera adequadas e, utilizando a funcionalidade de construção manual de rotas, corrigir os aspetos que desejar.

#### **4.4.5.2.5. Comparar resultados**

A última funcionalidade que se apresenta é a que permite comparar os resultados dos algoritmos para um determinado espaço de tempo. Esta foi considerada importante no que ao apoio à tomada de decisão diz respeito, uma vez que permite uma análise global ao funcionamento dos algoritmos para o conjunto de *inputs* submetidos, que com certeza traduzirão a realidade da empresa relativamente ao padrão de encomendas habitual. Apresenta-se de seguida o ecrã que traduz esta funcionalidade.



**Figura 28 – Ecrã representativo da funcionalidade Comparação de Algoritmos**

Como se pode ver na Figura 28, o ecrã de comparação de resultados mostra, para cada dia com simulação de rotas automáticas e para cada algoritmo, o número de veículos utilizados, a distância total percorrida no conjunto das rotas e a estimativa de consumo de combustível. No final é apresentado o somatório de cada um destes valores, para cada um dos algoritmos.

Da análise do caso dos dias acima pode-se aferir, tal como já se depreendia da análise de resultados realizada anteriormente, que por norma o *Algoritmo 1* necessita de menos veículos para realizar a entrega das encomendas porém, o *Algoritmo 2* serve as mesmas encomendas em menos quilómetros e, logo, com um consumo estimado de combustível inferior.

## 5. Conclusão

---

O objetivo desta dissertação era o de construir uma ferramenta de apoio à decisão para VRP. Após a conclusão do projeto acredita-se, tal como no início se acreditava, que a pertinência que levou à perseguição deste objetivo é proporcional aos custos logísticos associados à produção e distribuição de bens, e aumenta com o crescimento quase diário dos custos de combustível. A prova desta pertinência surge em vários quadrantes, de onde se destacam concursos para sugestão de soluções que permitam a redução de gastos de combustível, como o lançado pela transportadora RETA, com prémios monetários significativos (Anexo 4).

Apesar de o objetivo ser o referido, para que se pudesse almejar cumpri-lo, foi necessário um estudo aprofundado prévio sobre o que são os VRP, quais os seus constituintes, quais os objetivos genéricos, quais as características das principais instâncias já estudadas e quais as formas de resolução que bibliograficamente se apontam. Paralelamente a este estudo, para dotar a ferramenta desenvolvida de características tecnológicas de última geração, foi imprescindível a análise do que é possível fazer-se hoje em dia ao nível da representação cartográfica digital em aplicações próprias, tendo-se para tal estudado e aprendido a manipular a API da ferramenta *Google Maps*.

Analisando o SIG apoiado na Web, desenvolvido para o auxílio de decisores humanos no âmbito dos VRP, é possível e relevante afirmar-se que o mesmo potenciou todo o estudo anterior uma vez que, além da utilidade real da ferramenta desenvolvida, a mesma permitiu refletir, analisar e visualizar em termos práticos e minuciosos a investigação anteriormente realizada. Este facto manteve o interesse e a motivação sobre o desenvolvimento da ferramenta e sobre as suas potencialidades a alto nível, ao ponto de ser possível planear e imaginar diversos pontos de trabalho futuro.

Considera-se assim que a ferramenta construída pode ser útil no âmbito do objetivo para o qual foi criada: apoiar a formulação de decisões no âmbito de problemas reais de encaminhamento de veículos.

## **5.1. Trabalho futuro**

Conforme anteriormente referido, este projeto, dada a sua pertinência, não está concluído nem se consegue vislumbrar um fim. Tal acontece porque o mesmo, dada a preocupação com a escalabilidade com que foi construído, pode ser usado como base para projetos mais concretos, desenvolvidos para a especificidade de cada empresa, tendo em conta, nomeadamente, as restrições específicas de cada serviço de distribuição ou recolha. Como facilmente se observa pela arquitetura da aplicação, não é difícil incrementar ou substituir um algoritmo de criação de rotas por um que se ajuste melhor às características próprias de cada empresa/problema, ou aumentar uma nova funcionalidade ao sistema.

Em termos académicos também seria interessante avançar para o desenvolvimento e implementação de outros algoritmos exatos, heurísticas e meta-heurísticas que possibilitassem a comparação das soluções com as existentes e a visualização cartográfica das soluções emanadas por cada algoritmo.

## Bibliografia

---

- Archetti, C., Mansini, R., & Speranza, M. G. (2003). Complexity and Reducibility of the Skip Delivery.
- Aronoff, S. (1989). Geographic Information Systems: a management perspective. Ottawa, Canada: WDL Publications.
- Auren & NEO. (Março de 2007). Obtido em Fevereiro-Março de 2012, de VRP Web: <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html?VRP-Intro.html>
- Bard, J. F., Huang, L., Dror, M., & Jaillet, P. (1998). A branch and cut algorithm for the VRP with satellite facilities.
- Bolduc, M.-C., Laporte, G., Renaud, J., & Boctor, F. F. (2010). A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars. In *European Journal of Operational Research* (Vol. 202, pp. 122-130). Elsevier.
- Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. In *Computers & Operations Research* (pp. 140-151). Elsevier.
- Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. In *Transportation Science* (Vol. 39, pp. 104-118).
- Carlsson, J., Ge, D., Subramaniam, A., Wu, A., & Ye, Y. (1 de Maio de 2007). Solving Min-Max Multi-Depot Vehicle Routing Problem.
- Carravilla, M. A., & Oliveira, J. F. (2010). Programação Inteira - Resolução por Branch and Bound.
- Clarke, G., & Wright, J. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*.
- Coene, S., Arnout, A., & Spieksma, C. R. (2008). The Periodic Vehicle Routing Problem: A Case Study. Leuven - Belgium.
- Comissão Europeia. (25 de Abril de 2008). *A logística do transporte de mercadorias na Europa*. Obtido em 3 de Fevereiro de 2012, de Europa - Sínteses da legislação da UE: [http://europa.eu/legislation\\_summaries/environment/tackling\\_climate\\_change/l24456\\_pt.htm](http://europa.eu/legislation_summaries/environment/tackling_climate_change/l24456_pt.htm)



- Cordeau, Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. In *Journal of the Operational Research Society* (pp. 512-522).
- Cordeau, J.-F., Gendreau, M., Hertz, A., Laporte, G., & Sormany, J.-S. (2005). New heuristics for the vehicle routing problem. In A. Langevin, & D. Riopel (Edits.), *Logistics Systems: Design and Optimization*. Gerard.
- Dantzig, G., & Ramser, J. H. (Outubro de 1959). The Truck Dispatching Problem.
- Davis, C., & Neto, G. C. (2001). Arquitetura de Sistemas de Informação Geográfica.
- Department of the Environment (DoE). (1987). Handling Geographic Information. London.
- Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., & Soumis, F. (2002). VRP with Pickup and Delivery. In S. M. Applications, P. Toth, & D. Vigo (Edits.), *The Vehicle Routing Problem* (Vol. 9).
- Dubey, R. (2010). A Review on the Evolution of Vehicle Routing Problems.
- Dueker, K. J. (1979). Geo-Processing 1. In *Land resource information systems: a review of fifteen years experience* (pp. 105-128).
- Fischetti, M., Toth, P., & Vigo, D. (1994). Branch-and-Bound algorithm for the capacitated vehicle routing problem on directed graphs.
- Gendreau, M., Laporte, G., & Potvin, J.-Y. (2002). Metaheuristics for the Capacitated VRP. In S. -M. Applications, P. Toth, & D. Vigo (Edits.), *The Vehicle Routing Problem*.
- Gupta, R., Singh, B., & Pandey, D. (2010). Multi-Objective Fuzzy Vehicle Routing Problem - A Case Study.
- Holland, J. H. (1975). Adaptation in Natural and Artificial Systems.
- Jacobs-Blecha, C., & Goetschalckx, M. (1998). The Vehicle Routing Problem with Backhauls.
- Jozefowicz, N., Semet, F., & Talbi, E.-G. (2002). Parallel and Hybrid Models for Multi-objective Optimization: Application to the Vehicle Routing Problem.
- Laporte, G., & Louveaux, F. V. (1997). Solving Stochastic Routing Problems with the Integer L-Shaped Method. Canada - Belgium.
- Laporte, G., & Semet, F. (2002). Classical Heuristics for the Capacitated VRP. In S. -M. Applications, P. Toth, & D. Vigo (Edits.), *The Vehicle Routing Problem*.
- Laporte, G., Gendreau, M., Potvin, J.-Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. In *International Transactions in Operational Research* (pp. 285-300). Pergamon.

- Laporte, Nobert, & Taillefer. (1987). A branch-and-bound algorithm for the asymmetrical distance-constrained vehicle routing problem. Canadá.
- Lima, R. d., Lima, J. P., & Silva, T. V. (Abril de 2012). Roteirização em arcos com um sistema de informações geográficas para transportes: aplicação em coleta de resíduos sólidos urbanos.
- Lin, S.-W., Chou, S.-T., Lee, C.-Y., & Lee, Z.-J. (Março de 2012). A Hybrid Approach for Vehicle Routing Problem with Time Windows. World Science Publisher.
- Lysgaard, J. (1997). Clarke & Wright's Savings Algorithm.
- Malczewski, J. (1999). GIS and Multicriteria Decision Analysis. U.S.A. & Canada.
- Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. In *Annals of Operations Research* (pp. 513-623). J. C. Baltzer AG, Science Publishers.
- Parker, H. D. (1988). The unique qualities of a geographic information system: a comentary. In *Photogrammetric Engineering and Remote Sensing 54 (11)* (pp. 1547-1549).
- Ralphs, T., Hartman, J., & Galati, M. (2001). Capacitated Vehicle Routing and Some Related Problems.
- Renaud, J., & Boctor, F. F. (2002). A sweep-based algorithm for the fleet size and mix vehicle routing problem. Canada.
- Righini, G. (2000). Approximation algorithms for the Vehicle Routing Problem with pick-up and delivery.
- Taillard, É. D., Gambardella, L. M., Gendreau, M., & Potvin, J.-Y. (2001). Adaptive memory programming: A unified view os metaheuristics.
- Tan, C. C., & Beasley, J. E. (1984). A Heuristic Algorithm for the Period Vehicle Routing Problem. London, UK.
- Teixeira, R. G., & Cunha, C. B. (2002). Heurísticas para o problema de dimensionamento e roteirização de uma frota heterogênea utilizando o algoritmo out-of-kilter.
- Toth, P., & Vigo, D. (2002). An Overview of Vehicle Routing Problems. In S. M. Applications, P. Toth, & D. Vigo (Edits.), *The Vehicle Routing Problem* (pp. 1-23).
- Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2012). Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis.
- Yu, B., Yang, Z.-Z., & Xie, J.-X. (2011). A parallel improved ant colony optimization for multi-depot vehicle routing problem.

Zhang, Z., Qin, H., Lim, A., & Guo, S. (2010). Branch and Bound Algorithm for a Single Vehicle Routing Problem with Toll-by-Weight Scheme. Cordoba, Spain.

## **ANEXOS**

---

## ANEXO 1 – Pseudo-código do Algoritmo 1

```

Dia_Rotas = Dia para criar rotas
N_Encomendas=Número de Encomendas para Dia_Rotas
N_Veiculos=Número de Veículos total
N_Depositos=Número de Depósitos total
Array_depositos_sem_veiculos = []
Enquanto (N_Encomendas > 0)
    E = Encomenda que ainda não pertence a nenhuma rota (cujo cliente é C)
    Se(Tamanho(Array_depositos_sem_veiculos)!=N_Depositos)
        Encomenda E passa para o dia seguinte
    Senão
        V=NULL;
        Enquanto (Tamanho(Array_depositos_sem_veiculos)<N_Depositos) E (V=NULL)
            D=Depósito mais próximo de C que não pertença ao Array_depositos_sem_veiculos
            Enquanto não forem verificados todos os veículos do depósito D
                Se veículo V não está em nenhuma rota
                    Seleciona veículo V
            Se(V==NULL)
                D->Array_depositos_sem_veiculos
            Se(V!=NULL)
                Ponto1=Depósito D
                Ponto2= Cliente mais próximo do depósito D, com encomenda para Dia_Rotas
                Capacidade=V.Capacidade;
                Tempo=Tempo de percurso entre Ponto 1 e Ponto 2;
                l=0;
                Insere Ponto 1 na rota na ordem de visita l
                Array_Clientes_Encomenda_Ultapassa_Capacidade=[]
                Enquanto (Tempo+Tempo de percurso entre Ponto2 e Depósito D <= 8horas) e (Capacidade>0) e (N_Encomendas>0)
                    l++;
                    Insere Ponto 2 na rota na ordem de visita l para o veículo V
                    N_Encomendas--;
                    Se(N_Encomendas>0)
                        Ponto 1=Ponto2
                        Ponto 2=Cliente com encomenda (E) à menor distância de Ponto 1;
                        Tempo_Válido=True
                        Se (Capacidade-E.Peso<0)
                            Ponto2 -> Array_Clientes_Encomenda_Ultapassa_Capacidade
                            Enquanto (N_Encomendas-Tamanho(Array_Clientes_Encomenda_Ultapassa_Capacidade)>0) E (Tempo_Válido=True) E ((Encomenda Encontrada)
                                Ponto 2=Cliente com encomenda (E) a menor distância de Ponto 1 que não está no Array de clientes cuja encomenda ultrapassa o limite de capacidade;
                                Se(Capacidade - E.peso >= 0)
                                    Se(Tempo+Tempo de percurso entre Ponto1 e Ponto2+Tempo de percurso entre Ponto2 e Depósito D <= 8horas)
                                        Encomenda encontrada
                                        Tempo=Tempo+Tempo de percurso entre Ponto 1 e Ponto2;
                                        Capacidade=Capacidade-E.peso;
                                    Senão
                                        Tempo_Válido=False;
                                        Tempo=8horas;
                            Senão
                                Ponto2 -> Array_Clientes_Encomenda_Ultapassa_Capacidade
                            Se((Tempo+Tempo de percurso entre Ponto1 e Ponto2+Tempo de percurso entre Ponto2 e Depósito D <= 8horas)
                                Tempo=Tempo+Tempo de percurso entre Ponto 1 e Ponto2;
                                Capacidade=Capacidade-Peso da Encomenda;
                            Insere Depósito na rota na ordem l+1;

```

## ANEXO 2 – Pseudo-código do Algoritmo 2

```

//após os clusters formados fazer
Dia_Rotas = Dia para criar rotas
Para todos os depósitos D fazer
    i=0;
    Total_encomendas=nº total de encomendas no cluster D;
    Peso_Total_Encomendas=peso total das encomendas do cluster D
    Veic=Array dos veículos de D ordenados pela capacidade
    Enquanto (Total_encomendas>0)
        Ponto1=Depósito D
        Ponto2= Cliente ainda não atendido do cluster D, mais próximo de Ponto1, com encomenda (E) para Dia_Rotas
        Se (>=tamanho(Veic))
            E.data=E.data+1;
            Total_encomendas--;
        Senão
            Enquanto (Peso_total_encomendas<Veic[i].capacidade)
                i++;
            Veic_rotas=Veic[i]
            Capacidade=Veic_rotas.Capacidade-E.peso;
            Tempo=Tempo de percurso entre Ponto 1 e Ponto 2;
            j=0;
            Insere Ponto 1 na rota, na ordem de visita j, do veículo Veic_rotas
            Array_Clientes_Encomenda_Ultapassa_Capacidade=[]
            Enquanto (Tempo+Tempo de percurso entre Ponto2 e Depósito D <= 8horas) e (Capacidade>0) e (Total_encomendas>0)
                j++;
                Insere Ponto 2 na rota na ordem de visita j para o veículo Veic_rotas
                Total_encomendas--;
                Peso_total_encomendas-=E.peso
                Se(Total_encomendas>0)
                    Ponto 1=Ponto2
                    Tempo_Válido=True
                    Se (Capacidade-E.Peso<0)
                        Ponto2-> Array_Clientes_Encomenda_Ultapassa_Capacidade
                        Enquanto (Total_encomendas-Tamanho(Array_Clientes_Encomenda_Ultapassa_Capacidade)>0) E (Tempo_Válido=True) E ((Encomenda Encontrada)
                            Ponto 2=Cliente do cluster D com encomenda (E) a menor distância de Ponto 1, que não está no Array de clientes cuja encomenda ultrapassa o limite de capacidade
                            Se(Capacidade - E.peso >= 0)
                                Se(Tempo+Tempo de percurso entre Ponto1 e Ponto2+Tempo de percurso entre Ponto2 e Depósito D <= 8horas
                                    Encomenda encontrada
                                    Tempo=Tempo+Tempo de percurso entre Ponto 1 e Ponto2;
                                    Capacidade=Capacidade-E.peso;
                                    Senão
                                        Tempo_Válido=False;
                                        Tempo=8horas;
                                    Senão
                                        Ponto2-> Array_Clientes_Encomenda_Ultapassa_Capacidade
                                Se(Tempo+Tempo de percurso entre Ponto1 e Ponto2+Tempo de percurso entre Ponto2 e Depósito D <= 8horas
                                    Tempo=Tempo+Tempo de percurso entre Ponto 1 e Ponto2;
                                    Capacidade=Capacidade-E.peso;
                                Insere Depósito na rota na ordem j+1;
                                i++;
                            //Se o nº de encomendas encher mais do q um veículo, só vão ser analisados os veículos de capacidade igual ou abaixo
                            //Selecione veículo apropriado em termos de capacidade
                            //Se não se puder selecionar essa encomenda tenta-se outra
                            //Depois de cada encomenda introduzida, se ainda houver encomendas, prepara-se a introdução da próxima
                            //Se não se puder selecionar essa encomenda tenta-se outra
                            //fecha o ciclo interior
                            //fecha o ciclo exterior

```


### ANEXO 3 – Pseudo-código do Algoritmo de Pesquisa Local

```
para todas as rotas fazer
  Rota=Id da Rota em análise;
  Distancia_VMP= Distância para realização da rota utilizando algoritmo do vizinho mais próximo
  Distancia_minima=Distancia_VMP
  Ordem_rota=ler pontos para array por ordem de visita;
  Tam_Rota = tamanho do array Ordem_rota;
  melhor_rota_encontrada=0;
  nova_iteração=1;
  enquanto (nova_iteração==1)
    nova_iteração=0;
    i=0;
    enquanto (i<tam_rota-1)
      j=0;
      enquanto(j<i)
        Ordem_rota_aux[j]=Ordem_rota[j];
        j++;
        Ordem_rota_aux[j]=Ordem_rota[j+1];
        Ordem_rota_aux[j+1]=Ordem_rota[j];
        j+=2;
        enquanto(j<tam_rota)
          Ordem_rota_aux[j]=Ordem_rota[j];
          j++;
          Distancia_rota_aux=cálculo da distância realizada pela visita aos pontos Ordem_rota_aux pela ordem ditada por esse array
          Se(Distancia_rota_aux<Distancia_minima)
            Distancia_minima=Distancia_rota_aux;
            nova_iteração=1;
            array_nova_iteração=Ordem_rota_aux;
            i++;
          Se(nova_iteração=1)
            ordem_rota=array_nova_iteração;
            melhor_rota_encontrada=1;
          se(melhor_rota_encontrada==1)
            Alterar ordem da rota Rota na BDs para ordem_rota
            Alterar distancia da rota Rota na BDs para distancia_minima
```

## ANEXO 4 – Challenge InoCrowd – Redução de consumos de combustível

21/10/12 InoCrowd.com :: Challenge Details

ABOUT CONTACT US




find your solution NEWS

HOME CHALLENGES DASHBOARD

Welcome, João Casal | LOGOUT

> View Challenge



### Fuel consumption reduction for heavyweight vehicles (Tractor)

Type Practical Challenge  
Areas Business \Logistics, Energy \Energy efficiency, Engineering \Chemical engineering  
Award € 20.000,00 Deadline 2012-11-05  
Posted by Reta

#### Description

A technological solution is required to allow for a minimum reduction of 25% on the fuel consumption of the fleet of heavyweight vehicles from RETA and its clients. RETA is a company that belongs to "Grupo Luís Simões" and is leader in Portugal in the rental of semi-trailers.

[in Share](#)

#### Details

*(See attached file for the portuguese version | Versão portuguesa em anexo)*

##### Challenge details

RETA operates in the rental, maintenance and repair of heavyweight vehicles that are used to transport products and goods. It is a leader in Portugal in the rental of semi-trailers. RETA is the result of a merger between two companies – Socar (maintenance and repair) and Reta – Rent-a-cargo (rental), founded in 1989 and 1990 respectively. With over 20 years of experience, RETA has been positioning itself on the road transportation market by the quality of its products, the delivery of an innovative service and the quality of an increasingly diversified offer. RETA's strategy is guided by continuous innovation, constantly looking for the most efficient and adequate solutions to its clients in order to bring more value-add to the services provided.

RETA has a rental fleet of around 650 heavyweight vehicles, between tractors and semi-trailers. It also holds 3 repair shops in Portugal and provide maintenance and repair services to over 2.500 vehicles from its clients.

The surge on the fuel prices that has been recently observed, lead to a significant increase on the operating costs of the heavyweight vehicles, thus the company decided to post a challenge on InoCrowd in order to receive technological solutions that are able to reduce the fuel consumption of the vehicles to a minimum of 25%.

##### Solutions

The presented solutions need to be structured according to the following:

- Executive Summary;
- Detailed description of the proposed solution;
- Detailed description of the technology behind the solution;
- Detailed description of the reasons why the proposed solution is different to what the market currently offers; and
- Description of how the solution works (eventually still in the prototyping stage).

The proposed solutions need to consider and allow tests to be performed on RETA's or LSLI's vehicles.

The evaluation of the solutions will be performed through real-life tests where the solution will be applied to a set of vehicles (between 2 and 5) that will be driven along certain tracks that will allow for the consumptions to be compared before and after. The minimum mileage of each vehicle, for evaluation purposes should be 40.000 kms.

The selected solver(s) shall/can follow the evaluation tests. The proposed solutions should not contain technical or other type of data that is considered by the solver as "intellectual property" that cannot be shared.

All solvers that submit a solution will be notified about the result of the challenge. However, no detailed information will be given about the evaluation of the solutions.

##### Award

The seeker reserves the right to award the prize to more than one solution. In this situation the monetary prize will be distributed to all the winning

[https://inocrowd.com/lc\\_Presentation/Challenges.aspx](https://inocrowd.com/lc_Presentation/Challenges.aspx)



solutions.



NEWSLETTER

Subscribe ▶



USEFUL LINKS

[Privacy policy](#)

[Terms of use](#)

FAQ

[How can I join?](#)

[What is a challenge?](#)

[How are challenges evaluated?](#)

[How secure is the information posted at InoCrowd?](#)

DEVELOPED BY SIMETRICIRCLE