

Optimização de Rotas na Recolha de Resíduos Urbanos

Modelos e Algoritmos

Maria Cândida Vergueiro Monteiro Cidade Mourão

*Dissertação elaborada para provas de doutoramento em
Matemática Aplicada à Economia e Gestão no
Instituto Superior de Economia e Gestão da
Universidade Técnica de Lisboa
orientada pela Professora Doutora Maria Teresa Nunes Chaves de Almeida*

(edição revista) Janeiro de 1999

Lisboa

Índice

À Minha Querida "Famelga"

1. Introdução	1
2. Problemas de Otimização de Forma com Fronteiras Livres	2
2.1. Introdução	2
2.2. Problemas do Círculo Clássico	3
2.3. Problemas com Fronteiras Livres	7
2.4. Problemas com Restrições de Capacidade	10
2.4.1. Introdução	10
2.4.2. Métodos Clássicos	12
2.4.3. Minimax	13
2.5. Outros Problemas	15
2.6. Aplicações a Simulacres Reais	17
3. O Problema de Encaixe de Peças Sólidas (PPES)	19
3.1. Definições e Notação	21
3.2. Formulação do Problema de Otimização	23
4. Algoritmos para o PPES	24
4.1. Introdução	24
4.2. Integração com Algoritmos de Restrições de Capacidade e de Capacidade	25
4.3. Algoritmo para a Determinação de Peças	26
4.4. Resultados com Algoritmos de Restrições de Capacidade e Capacidade	27
4.4.1. Introdução	27

Índice

Resumo	XI
1. Introdução	1
2. Problemas de Optimização de Rotas com Procura nos Arcos	
2.1. Introdução	5
2.2. Problema do Carteiro Chinês	7
2.3. Problemas com Procura nos Arcos	9
2.4. Problemas com Restrições de Capacidade	
2.4.1. Introdução	10
2.4.2. Métodos Heurísticos	12
2.4.3. Minorantes	13
2.5. Outros Problemas	15
2.6. Aplicações a Situações Reais	17
2.7. O Problema de Recolha de Resíduos Sólidos (PRRS)	
2.7.1. Definições e Notação	21
2.7.2. Formalização em Programação Linear Inteira	25
3. Relaxações para o PRRS	
3.1. Introdução	31
3.2. Relaxação com Inclusão de Restrições de Equilíbrio e de Capacidade	33
3.3. Algoritmo para a Determinação de Minorantes	51
3.4. Relaxações com Inclusão de Restrições de Equilíbrio, Capacidade e Conexidade	
3.4.1. Introdução	55

3.4.2.	Super-Vértices	59
3.4.3.	Duplicação de Destinos	67
3.5.	Comparações entre os Minorantes	76
4.	Heurísticas para o PRRS	
4.1.	Introdução	87
4.2.	Descrição Geral das Heurísticas Construtivas	89
4.3.	Geração de Subrotas	
4.3.1.	Introdução	93
4.3.2.	Método de Geração por Partição de um Circuito Euleriano	94
4.3.3.	Método de Geração por Procuras	98
4.3.4.	Exemplo	101
4.4.	Agregação Sequencial de Subrotas	106
Exemplo		110
4.5.	Agregação de Subrotas por Proximidade	
4.5.1.	Introdução	113
4.5.2.	Multigrafo de Subrotas	
Definições		115
Algoritmo para Construção do Multigrafo de Subrotas		119
Exemplo		121
4.5.3.	Método de Emparelhamentos	
Descrição do Método		122
Exemplo		125
4.5.4.	Método de Sementes	
Descrição do Método		127
Exemplo		131
4.6.	Geração de uma Solução Admissível	134
4.7.	Análise Teórica da Heurística H1	137
4.8.	Método Melhorativo	
4.8.1.	Introdução	141
4.8.2.	Descrição do Método	
Introdução		142
Trocas de Arcos numa Subrota		143
Trocas de Arcos entre Subrotas Adjacentes		147
4.8.3.	Exemplo	151
5.	Resultados Computacionais	
5.1.	Introdução	153
5.2.	Problemas Teste em Redes Pouco Densas	
5.2.1.	Geração	155
5.2.2.	Resultados Computacionais	156

5.3. Problemas Teste em Redes Esparsas e Planares	
5.3.1. Geração	160
Geração de Grafos Não Orientados, Planares e	
Conexos	160
Transformação num Grafo Orientado, Fortemente	
Conexo e Planar	165
Geração dos Parâmetros	168
5.3.2. Resultados Computacionais	
Resultados Computacionais de H1 em problemas T2	174
Resultados Computacionais de H2 em problemas T2	176
Resultados Computacionais de H3 em problemas T2	178
Comparação entre as Heurísticas H1, H2 e H3	180
6. Conclusões	183
Anexos	
A1. Identificação dos Casos Possíveis para o Vértice 1	189
A2. Tabelas de Resultados Computacionais para Problemas T1	193
A3. Tabelas de Resultados Computacionais para Problemas T2	197
Listas	
L1. Lista de Algoritmos	203
L2. Lista de Exemplos	204
L3. Lista de Figuras	205
L4. Lista de Fluxogramas	206
L5. Lista de Tabelas	206
Referências Bibliográficas	207

Resumo

Os problemas de determinação de rotas óptimas para um ou mais veículos são, em geral, classificados em dois grandes grupos: problemas com procura nos vértices ("Node Routing Problems" - NRP) e problemas com procura nos arcos ("Arc Routing Problems" - ARP). A inclusão, nestes problemas, de restrições quanto às capacidades dos veículos conduzem, por um lado, a um VRP ("Vehicle Routing Problem") e, por outro, a um CARP ("Capacitated Arc Routing Problem"). Muitos exemplos podem ser dados de aplicações reais deste tipo de problemas, entre os quais a recolha de resíduos sólidos. Tratando-se, em geral, de problemas de "difícil" resolução, torna-se importante o desenvolvimento de "bons" métodos aproximativos. Os problemas com procura nos vértices (com e sem restrições adicionais) têm, contudo, despertado mais atenções que os de procura nos arcos.

Este trabalho tem como objectivo o estudo de um problema, denominado por PRRS (Problema de Recolha de Resíduos Sólidos), que pode ser visto como um CARP com restrições adicionais. O PRRS baseou-se no caso da determinação de rotas para os veículos afectos à recolha de resíduos sólidos na cidade de Lisboa.

Após formalizar o PRRS desenvolvem-se métodos aproximativos. Três relaxações da formalização apresentada fornecem três minorantes válidos para o valor óptimo do PRRS. Duas destas relaxações são, como se prova, resolúveis por problemas de transporte, enquanto a terceira pode ser resolvida por um problema de fluxo de custo mínimo. São estabelecidas algumas desigualdades entre os valores destas relaxações em certas instâncias dos problemas.

Com o objectivo de obter “boas” soluções admissíveis são desenvolvidas três heurísticas construtivas e uma melhorativa.

Alguns dos métodos foram codificados em Pascal e testados num conjunto de problemas teste gerados aleatoriamente. Como se mostra, os resultados quer em termos de valores percentuais dos desvios relativos, quer em termos de estrutura das soluções admissíveis podem ser considerados bastante razoáveis.

Palavras Chave

Problemas de Rotas com Procura nos Arcos; Formalização; Minorante; Relaxação; Majorante; Heurística.

Agradecimentos

Agradecer à minha orientadora Professora Doutora Teresa Almeida pela supervisão desta tese.

Um obrigada bem grande fica para os meus amigos e colegas que me aturaram e me ampararam, para que eu não caísse e não ficasse pelo caminho. Especial destaque merecem, sem dúvida, o Paulo e o Onofre. Agradeço ao Carlão e ao Zeca por, num tão curto prazo, terem feito a capa.

O maior dos agradecimentos devo-o à “Instituição Avós” por ter funcionado, sem qualquer reclamação, a 300%. Ao Carlos agradeço o constante apoio e incentivo e aos meus filhotes o facto de existirem. Um especial obrigada também para os meus irmãos e Pais.

Agradeço ainda ao Instituto Superior de Economia e Gestão e ao Centro de Investigação Operacional os apoios para a realização desta tese, nomeadamente no que diz respeito à participação em conferências. À Junta Nacional de Investigação Científica agradeço a comparticipação nas despesas de publicação da tese.

Imperfeições ou quaisquer erros que possam existir nesta tese são, obviamente, de minha exclusiva responsabilidade.

Simbologia

- $|A|$ cardinal do conjunto A ;
- $\lceil x \rceil$ menor inteiro superior ou igual ao número real x ;
- $x \text{ div } y$ divisão inteira de x por y ;
- $x \in [a \mp b] \Leftrightarrow a - b \leq x \leq a + b$;
- z^* valor óptimo de um problema em que se pretende *Min* z ;
- \underline{z} minorante para o valor óptimo z^* de um problema ($\underline{z} \leq z^*$);
- \bar{z} majorante para o valor óptimo z^* de um problema ($\bar{z} \geq z^*$);
- c.c. caso contrário;
- fim de prova;
- ◆ fim de exemplo.

Abreviaturas

- ARP Problema de determinação de rotas óptimas para veículos, com procura nos arcos ("Arc Routing Problem");
- CARP ARP para veículos de capacidade limitada ("Capacitated Arc Routing Problem");
- CCPP Problema do carteiro Chinês com restrições de capacidade limitada ("Capacitated Chinese Postman Problem");
- CE Circuito Euleriano;
- CPP Problema do carteiro Chinês ("Chinese Postman Problem");
- ETRS Estação de Tratamento de Resíduos Sólidos;
- NRP Problemas de determinação de rotas óptimas para veículos, com procura nos nodos ("Node Routing Problem");
- PRRS Problema de Recolha de Resíduos Sólidos;
- PT Problema de Transporte;
- RPP Problema do carteiro rural ("Rural Postman Problem");
- VRP NRP para veículos de capacidade limitada ("Vehicle Routing Problem").

Optimização de Rotas na Recolha de Resíduos Urbanos

Modelos e Algoritmos

Capítulo 1. Introdução

Os problemas de determinação de rotas óptimas para um ou mais veículos são, de geral, classificados em dois grandes grupos: problemas com pontos nos vértices (VRP - *Vehicle Routing Problem*) e problemas com pontos em arco (ARP - *Arc Routing Problem*).

A título ilustrativo, considero-se o problema de determinação de rotas para os veículos que efectuam a recolha de resíduos sólidos em Lisboa. De acordo com dados fornecidos pelo Departamento de Higiene Urbana e Resíduos Sólidos da Câmara Municipal de Lisboa (CML), em algumas zonas de Lisboa podem ser identificados dois tipos distintos de problemas associados a dois tipos distintos de veículos de recolha. Por um lado, a CML dispõe de um certo número e tipo de veículos para a recolha de resíduos que são depositados em contentores de pequena capacidade (120 l ou 240 l) instalados individualmente por edifício. Esteve referido um problema com "pontos" nos arcos, pois em cada um desses locais existem contentores de geral um ou dois por edifício. Existem ainda contentores de maior capacidade (1100 l), cuja localização se pode considerar discreta, dada a distância que os separa. Desde a recolha, para cada edifício por veículos diferentes, pode ser considerado um problema distinto de arcos, com "pontos" nos vértices. Portanto, em termos de problemas, optimizar o percurso dos veículos, minimizando o custo (ou tempo) total de recolha.

Capítulo 1. Introdução

Os problemas de determinação de rotas óptimas para um ou mais veículos são, em geral, classificados em dois grandes grupos: problemas com procura nos vértices (NRP – *Node Routing Problems*) e problemas com procura nos arcos (ARP – *Arc Routing Problems*).

A título ilustrativo, considere-se o problema da determinação de rotas para os veículos que efectuem a recolha de resíduos sólidos em Lisboa. De acordo com dados fornecidos pelo Departamento de Higiene Urbana e Resíduos Sólidos da Câmara Municipal de Lisboa (CML), em algumas zonas de Lisboa, podem ser identificados dois tipos distintos de problemas associados a dois tipos distintos de veículos de recolha. Por um lado, a CML dispõe de um certo número e tipo de veículos para a remoção de resíduos que são depositados em contentores de pequenas capacidades (120ℓ ou 240ℓ) localizados individualmente por edifício. Está-se perante um problema com “procura” nos arcos, pois em cada rua existem muitos contentores (no geral um ou dois por edifício). Existem ainda, contentores de maior capacidade (1100ℓ), cuja localização se pode considerar discreta, dada a distância que os separa. Sendo a recolha, neste caso, efectuada por veículos diferentes, pode identificar-se um problema distinto do anterior, com “procura” nos vértices. Pretende-se, em ambos os problemas, otimizar o percurso dos veículos, minimizando o custo (ou tempo) total de recolha.

A par da recolha de resíduos sólidos, muitos outros exemplos podem ser dados de aplicações reais deste tipo de problemas: a leitura dos contadores domésticos de electricidade, gás ou água; a lavagem automática de ruas; a inspecção preventiva de condutas; a recolha de correspondência depositada pelos utentes em marcos de correio; a distribuição de correio ao domicílio; etc. Em todas estas aplicações é necessário introduzir restrições adicionais, relativas, por exemplo, às capacidades dos veículos, aos horários de trabalho dos funcionários, às normas que regulam o trânsito e o estacionamento.

A inclusão, em problemas do tipo descrito, de restrições quanto às capacidades dos veículos conduzem, por um lado, a um VRP (*Vehicle Routing Problem*) e, por outro a um CARP (*Capacitated Arc Routing Problem*). Golden e Wong [42] descrevem, comparam e classificam este tipo de problemas. Sendo, em geral, NP-difíceis, não se conhecem algoritmos que forneçam a sua solução óptima em tempo polinomial. Este facto faz com que seja importante o desenvolvimento de “bons” métodos aproximativos, justificando assim a atenção que têm tido através dos tempos. Porém, os problemas com procura nos vértices (com e sem restrições adicionais) têm despertado mais atenções que os com procura nos arcos. Em parte este facto justifica também a escolha do tema desta tese.

De notar que um dos problemas operacionais, mais difíceis e actuais, enfrentados pelas câmaras municipais de grandes cidades, é o da recolha de resíduos sólidos.

O presente trabalho visa o estudo do problema da determinação de rotas para os diferentes veículos afectos à recolha de resíduos sólidos em Lisboa, no caso de contentores de pequenas capacidades, como se descreve de seguida.

O problema da recolha de resíduos sólidos, doravante designado por PRRS, que esteve na origem deste trabalho, pode, como se verá, ser encarado como um CARP com restrições adicionais. De facto, os contentores de pequenas capacidades localizados em frente de cada edifício conduzem a um problema com procura nos arcos. Para além disto, a capacidade limitada dos veículos utilizados na recolha dos resíduos faz com que seja necessário introduzir restrições de capacidade no problema. Características específicas do problema em estudo fá-lo diferir do CARP, sendo necessário considerar condições não usuais para a sua formalização.

Todos os veículos que efectuem a remoção de resíduos sólidos na cidade de Lisboa partem da garagem, localizada na Av. 24 de Julho, levando apenas o motorista. A fim de completar a tripulação, dirigem-se ao Posto de Serviço da zona onde vão efectuar a recolha. Do Posto de Serviço de cada zona parte um certo número fixo de veículos para fazer a recolha de resíduos sólidos.

A rota de cada veículo é interrompida diversas vezes, de forma a que não seja excedida a sua capacidade. Em certos locais, previamente estabelecidos, os veículos dirigem-se à Estação de Tratamento de Resíduos Sólidos (ETRS) para despejar os resíduos recolhidos.

No fim da recolha, os veículos passam primeiro pelo Posto de Serviço da zona de onde partiram, para largar a tripulação, com excepção do motorista, deslocando-se de seguida à ETRS, a caminho da garagem. A figura 1.1 representa uma rota admissível para um veículo.

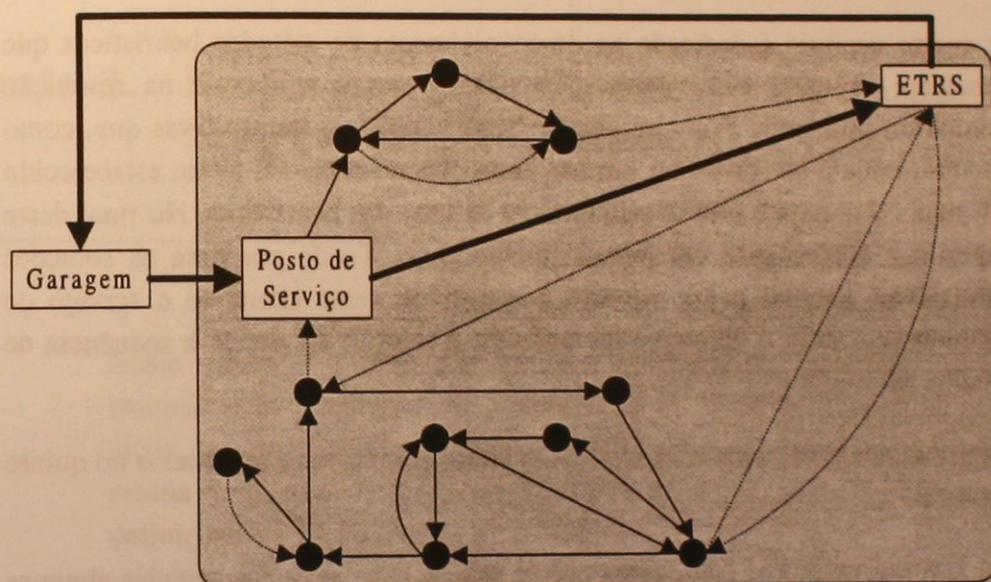


Figura 1.1: Representação de uma rota admissível para um veículo

As setas mais carregadas da figura envolvem apenas a determinação dos caminhos mais curtos entre os respectivos pontos. De facto, no início, no percurso entre a garagem e o Posto de Serviço não é efectuada recolha. O mesmo acontece nas duas viagens finais, do Posto de Serviço à ETRS e desta à garagem. Contudo, a parte sombreada representa um problema de “difícil” resolução, dado ser necessário ter em conta tanto a capacidade do veículo como as quantidades de resíduos a recolher em cada rua [42]. Assim, o problema em estudo nesta tese representa a parte sombreada da figura.

Inicia-se o segundo capítulo com um breve enquadramento histórico do problema. Introduzem-se de seguida os principais problemas com procuras nos arcos existentes, bem como alguns métodos propostos para as suas resoluções, quer exactas quer aproximadas. A formalização em programação linear inteira do PRRS é precedida da principal notação utilizada ao longo do texto.

Os minorantes desenvolvidos surgem, como se verá no terceiro capítulo, de relaxações da formalização feita. Para obter o primeiro minorante relaxam-se,

entre outras, as restrições de conectividade. Dois minorantes alternativos surgem na tentativa de incluir algumas das restrições de conectividade na primeira relaxação. A solução ótima de qualquer das três relaxações pode ser determinada por problemas de simples resolução. Assim, as duas primeiras são transformadas em problemas de transporte, enquanto a terceira é resolvida por um problema de fluxos de custo mínimo. Nesta tese prova-se que o valor ótimo de cada um destes problemas representa um minorante válido para o valor ótimo do problema inicial, sendo ainda estabelecidas algumas desigualdades entre eles.

O quarto capítulo é dedicado ao desenvolvimento de métodos heurísticos que fornecem soluções admissíveis, passíveis de serem aplicáveis na resolução prática do problema. Primeiro são descritas heurísticas construtivas que, como se verá, podem ser divididas em três fases fundamentais. É ainda estabelecido um majorante para o pior desvio relativo de uma das heurísticas. No final deste capítulo é apresentado um método melhorativo. Tendo por base as soluções admissíveis geradas pelos métodos construtivos, tenta alterar-se o serviço de algumas ruas com o objectivo de melhorar a solução ao alterar a sequência de recolha gerada.

Os principais resultados computacionais são apresentados e analisados no quinto capítulo.

Por fim são tiradas as conclusões que se julgam relevantes e apontadas algumas sugestões para trabalho futuro.

No primeiro anexo são descritos alguns casos que, embora podendo ocorrer na prática, não se desenvolvem no texto dado serem de estudo semelhante ao caso considerado na escrita desta tese. Tabelas com os resultados computacionais mais detalhados constam dos segundo e terceiro anexos.

Capítulo 2. Problemas de Optimização de Rotas com Procura nos Arcos

Neste capítulo é feito um breve enquadramento histórico e são resumidos os problemas de optimização de rotas com procura nos arcos, nos quais se pode incluir o PRRS problema que se apresenta e estuda nesta tese. A formalização do PRRS é precedida de algumas definições e notação utilizada.

2.1. Introdução

Leonhard Euler, em 1736, equacionou o primeiro problema de teoria dos grafos. O problema clássico das sete pontes sobre o rio Pregol em Königsberg (actual Kaliningrad), em que Euler se questionou sobre a possibilidade de atravessar as pontes passando uma e uma só vez em cada uma (Figura 2.1), foi o primeiro problema nos arcos que se conhece. Sendo possível representar as pontes por arestas que ligam os diversos nodos (Figura 2.2) representativos das margens do rio e ilhas, o problema pode ser definido num multigrafo não orientado. Consiste em determinar a cadeia (o caminho se o grafo for orientado) que inclua todas as arestas do grafo, sem repetições. Tendo sido identificado por Euler, tal cadeia (caminho) é hoje designada por cadeia (caminho) Euleriana.

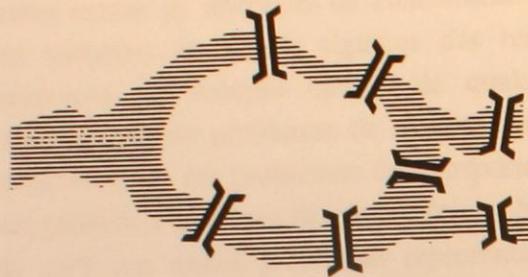


Figura 2.1: Pontes de Königsberg

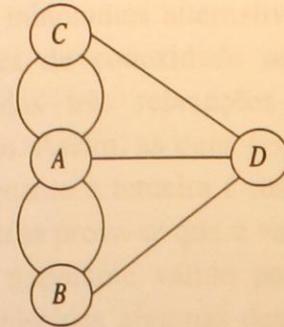


Figura 2.2: Multigrafo associado às pontes de Königsberg

Euler [36] mostrou que existe ciclo Euleriano (ou seja, cadeia Euleriana com início e fim no mesmo vértice), num grafo não orientado, se e só se todos os vértices tiverem grau par. Do século XVIII à actualidade, foram feitas provas das condições necessárias e suficientes para a existência de tais caminhos/cadeias (circuitos/ciclos) em grafos orientados, não orientados e mistos. Ford e Fulkerson [39] compilaram e demonstraram os resultados mais relevantes.

Numa visão macro pode considerar-se que os problemas de optimização de rotas têm como objectivo a determinação de um conjunto de circuitos, representando percursos a efectuar por uma frota de veículos, que minimize o custo total do percurso, respeitando certo tipo de restrições. Tipicamente estas restrições exigem a inclusão nos circuitos dos vértices de um certo conjunto de vértices e/ou dos arcos de um certo conjunto de arcos. Enquanto o primeiro tipo de restrições sugere que seja considerado um modelo nos vértices, como por exemplo o problema do caixeiro viajante (TSP – *Traveling Salesman Problem*), o segundo tipo de exigências conduz a modelizações nos arcos, como por exemplo o problema do carteiro Chinês (CPP – *Chinese Postman Problem*) ou o problema com procura nos arcos (ARP – *Arc Routing Problem*).

No TSP pretende identificar-se o circuito de custo mínimo que inclui uma e uma só vez cada um dos vértices de um grafo. O CPP consiste em determinar o circuito de custo mínimo que inclua todos os arcos de um grafo. O ARP difere do CPP por só se exigir que seja obrigatoriamente incluído um subconjunto do conjunto inicial de arcos. Os restantes arcos existentes podem ou não fazer parte do circuito final. Quando se pretende que sejam incluídos no circuito vértices e arcos, em simultâneo, define-se o GRP (*General Routing Problem*).

Os casos múltiplos como o k -TSP, o k -CPP e o k -GRP definem-se quando se pretende a determinação de k circuitos como os descritos, cada um dos quais incluindo ainda um vértice pré definido, usualmente designado por depósito. Estes problemas correspondem a modelizações de aplicações em que se

pretende determinar as rotas de uma frota constituída por k veículos, com $k \geq 2$, minimizando o custo total.

Abordagens mais aplicáveis a casos práticos surgem quando se considera que os vértices e/ou arcos têm procura e os veículos capacidade limitada. Assim, se as procuras estão em locais bem distintos definem-se problemas com procuras nos vértices, como por exemplo o VRP (*Vehicle Routing Problem*). Quando as procuras se podem considerar "contínuas" consideram-se nos arcos e surgem os problemas como o CCPP (*Capacitated Chinese Postman Problem*) e o CARP (*Capacitated Arc Routing Problem*), que difere do CCPP por existirem arcos de procura nula.

O problema em foco nesta tese, PRRS, pode ser visto como um CARP com restrições adicionais. Tratando-se de um problema com procura nos arcos vai fazer-se um breve resumo histórico incorporando em especial esta classe de problemas. Os artigos de Eiselt et al. [34], [35], o livro de Evans e Minieka [37] e o capítulo de Assad e Golden [2], no livro editado por Ball et al. [5], devem antes de mais ser referidos como as compilações mais recentes e, sem dúvida, mais importantes de resultados neste campo. Embora não tão recente, o artigo de Lenstra e Rinnooy Kan [54] é uma referência também importante, no respeitante a resumo bibliográfico de problemas com procuras nos arcos.

2.2. *Problema do Carteiro Chinês*

Como é óbvio, os caminhos Eulerianos estão intrinsecamente relacionados com o problema do carteiro Chinês (CPP), introduzido pelo matemático Chinês Meigu Guan [49], [50] (Mei-Ko Kwan) em 1962. O CPP foi introduzido por Guan com o objectivo de determinar o percurso a ser seguido por um carteiro que efectue a distribuição de correio a pé, numa cidade chinesa. Existindo, em cada rua, várias casas onde é necessário distribuir correio, a procura deve ser considerada contínua e logo deve ser modelado um problema com procura nos arcos.

Em 1962 Ford e Fulkerson [39] ⁽¹⁾ apresentaram condições necessárias e suficientes para a existência de um ciclo Euleriano num grafo misto.

Os principais resultados relacionando emparelhamentos, cadeias Eulerianas e o problema do carteiro Chinês obtidos, fundamentalmente, na década de sessenta

(1) A referência corresponde à 6ª edição, editada mais recentemente.

foram compilados em 1973 por Edmonds e Johnson [28]. Edmonds e Johnson [28] propuseram o primeiro algoritmo para determinar um circuito Euleriano num grafo Euleriano. Neste mesmo artigo é abordado o CPP em grafos orientados, não orientados e mistos e são apresentados algoritmos para determinar a solução óptima do CPP orientado, não orientado e de alguns casos particulares do CPP misto.

Edmonds e Johnson [28] provaram que as arestas adicionais num CPP não orientado podem ser obtidas com a resolução de um emparelhamento perfeito de custo mínimo entre os vértices de grau ímpar. Mostraram assim tratar-se de um problema resolúvel em tempo polinomial. Os autores formalizaram também o CPP em grafos orientados como um problema de fluxos e este problema passou assim a pertencer à classe dos tratáveis. Beltrami e Bodin [6] resolveram o CPP orientado recorrendo a um problema de transporte. Mais tarde, em 1988, Lin e Zhao [61] propuseram um algoritmo alternativo, de fluxos, para a resolução do CPP orientado que mantém a admissibilidade dual.

Edmonds e Johnson [28] apresentaram também uma heurística para o CPP misto baseada no método que resolve o mesmo problema se o grafo original for par, ou seja, se for apenas formado por vértices de grau par. Frederickson [40] identificou casos em que este método não providencia uma solução admissível mas, como mostrou, a solução obtida pode ser transformada numa admissível, com o mesmo valor. A dificuldade de resolução deste problema foi corroborada, em 1976, por Papadimitriou [73] ao provar que o CPP em grafos mistos é NP-difícil.

Em 1979, Minieka [66] formalizou o CPP misto como um problema de fluxos com ganhos numa rede expandida. Os algoritmos existentes para a resolução de problemas de fluxos com ganhos não garantem a integralidade da solução. Papadimitriou [73] provou que obter uma solução inteira num problema de fluxos com ganhos é um problema NP-difícil.

Mais recentemente, em 1993, Ralphs [79] também formalizou e demonstrou algumas propriedades para o CPP misto. O autor formalizou este problema como um problema de fluxos com um conjunto de restrições adicionais e complicativas. A relaxação obtida a partir desta formalização origina, segundo o próprio, bons minorantes para o CPP misto.

Em resumo, o CPP é NP-difícil [73], embora, como anteriormente frisado, tenha instâncias para as quais se conhecem algoritmos polinomiais. Pearn [77] apresenta estas instâncias referindo a quem se deve o primeiro algoritmo polinomial e a respectiva complexidade.

Algumas tentativas de resumos dos estudos desenvolvidos sobre o CPP foram feitas ao longo dos tempos. É disto exemplo o livro de Minieka, sendo a primeira edição de 1978, e a edição mais actual de 1992, revista por Evans e Minieka [37]. Mais recentemente, em 1995, foi publicado o artigo de Eiselt et al. [34] e o livro de Ball et al. [5] focando problemas com procura nos arcos, como é o caso do CPP. Estas publicações podem ser consideradas como as compilações mais recentes e importantes.

2.3. *Problemas com Procura nos Arcos*

Num problema com procura nos arcos, denominado por ARP (*Arc Routing Problem*), pretende determinar-se o circuito de custo mínimo que inclui um subconjunto de arcos pré definido.

Em diversos artigos o ARP é designado por RPP (*Rural Postman Problem*) dado ter sido definido, pela primeira vez em 1974 por Orloff [70], com o objectivo de determinar o conjunto de percursos a efectuar por um carteiro rural.

Sendo o CPP usualmente definido em grafos conexos, no ARP tal pode não se verificar. Ou seja, o grafo induzido pelo conjunto de arcos que têm que ser obrigatoriamente incluídos no circuito de custo mínimo pode ser desconexo. De facto, tratando-se da modelização do percurso a efectuar por um carteiro rural, este distribui o correio em diferentes zonas (vilas, lugares, casas isoladas, etc.), significativamente afastadas umas das outras. Neste caso, a resolução do ARP exige a transformação do grafo num grafo conexo, da melhor forma possível. Considerando cada componente conexas como um super-vértice, tornar o grafo conexo pode basear-se na identificação do caminho de menor custo que inclui cada um desses super-vértices uma única vez. Para além do CPP, em cada componente conexas, a resolução do ARP exige assim a "resolução" de um TSP para a ligação das diferentes componentes conexas. Logo, o ARP contém o TSP como caso particular. Tal justifica o facto do ARP representar, em geral, um problema de difícil resolução.

Orloff [72] referiu que a dificuldade de resolução do ARP aumenta com o aumento do número de componentes conexas, mesmo não sendo consideradas restrições de capacidade. Logo, o simples facto de passar de um CPP (onde todos os arcos têm que ser incluídos no circuito) para um ARP (onde apenas um conjunto restrito de arcos tem que ser obrigatoriamente incluído na solução) pode dificultar bastante a resolução do problema. Lenstra e Rinnooy Kan [53], em 1976, provaram que tanto o ARP orientado como o não orientado são

problemas NP-difíceis, sendo, lembre-se, o CPP orientado e o não orientado polinomiais.

Em 1982, Christofides et al. [20] mostraram ser possível transformar o TSP assimétrico num ARP orientado. Como afirmaram, passar de um TSP simétrico ao assimétrico é simples. Porém, no caso de problemas com procuras nas arestas não é trivial a transformação dos algoritmos existentes para o ARP não orientado ⁽¹⁾ em algoritmos para o ARP orientado ⁽²⁾ e vice-versa. Assim, propuseram dois algoritmos de partições e avaliações sucessivas, um para o ARP não orientado [19] e outro para o ARP orientado [20].

Assad e Golden [2] apresentaram formalizações existentes tanto para o caso orientado (de Ball e Magazine [4]) como para o caso não orientado (de Corberán e Sanchis [25]) e heurísticas propostas pelos mesmos autores.

2.4. Problemas com Restrições de Capacidade

2.4.1. Introdução

Introduzir restrições de capacidade nos problemas referidos é uma das generalizações mais pertinentes e vulgares. Tratando-se, em geral, de modelizações para recolha e distribuição de produtos, na maior parte de aplicações práticas há que considerar restrições de capacidade dos veículos, tempo de trabalho, ou seja, restrições que imponham limites nas rotas. Surgem então os problemas em que é necessário considerar mais de uma rota por veículo.

Assim, nos problemas com procuras nos arcos e restrições de capacidade, CCPP (*Capacitated Chinese Postman Problem*) / CARP (*Capacitated Arc Routing Problem*), pretende determinar-se o conjunto de circuitos de custo mínimo que incluem o depósito e todos os arcos de procura do grafo. Tal como

(1) Designado, por estes autores, por URPP – *Undirected Rural Postman Problem*.

(2) Designado por DRPP – *Directed Rural Postman Problem*.

anteriormente, no CCPP todos os arcos têm procura positiva, podendo no CARP existir arcos de procura nula.

Certos autores, perante aplicações com um número k ($k \geq 2$) pré estabelecido de veículos, definem as versões múltiplas dos problemas, ou seja o k -CPP. Em 1978, Frederickson et al. [41] provaram que o k -CPP é NP-difícil, quer em grafos orientados quer em grafos não orientados. Pearn [77] apresentou casos do k -CPP resolúveis em tempo polinomial, quer os já existentes, referindo o primeiro algoritmo polinomial conhecido, quer os identificados pelo próprio.

O CARP, tal como é hoje conhecido, só foi definido em 1981 por Golden e Wong [42], em grafos não orientados.

Para além do CARP, Golden e Wong [42] definiram a 0.5-aproximação do CCPP como sendo o problema de determinação de uma solução admissível para o CCPP de valor nunca superior a 1.5 do valor óptimo. Provaram que a 0.5-aproximação do CCPP (e do CARP) é NP-difícil. Identificando e descrevendo problemas relacionados, como o CPP, o ARP, o CCPP, o TSP, o VRP e o GRP, explicaram como todos estes podem ser encarados como casos particulares do CARP.

Assad et al. [3] apresentaram casos do CCPP resolúveis em tempo polinomial. Estes autores salientaram também que o simples passar de um CPP para um ARP pode dificultar bastante a resolução, podendo passar a obter-se problemas NP-difíceis. Logo os casos resolúveis para o CCPP nunca se podem estender ao CARP. Estes factos justificam igualmente a dificuldade de adaptação de métodos para o CCPP em métodos para o CARP.

Golden e Wong [42] formalizam o CARP, em redes não orientadas, definindo, contudo, variáveis orientadas. Eiselt et al. [35] apresentaram e comentaram tanto esta formalização como uma proposta por Belenguer e Benavent, em 1991, também para grafos não orientados, mas definindo variáveis não orientadas. Tratam-se das únicas formalizações conhecidas para o CARP ([35]).

Aplicações práticas do CARP são também em geral abordadas por uma perspectiva prática, não incluindo formalizações dos problemas originais, excepção feita ao artigo de Roy e Rousseau [82]. Estes autores formalizaram o problema do carteiro Canadano, por eles designado por CPC (*Canadian Post Corporation*), adaptando a formalização de Golden e Wong [42]. Na referência [82] supõe-se que cada rota tem um tempo limite, eventualmente diferente, e não se consideram as restrições de capacidade usuais. Trata-se também de um problema em redes não orientadas, dado o carteiro se movimentar a pé, sendo portanto indiferente o sentido da rua.

Considerando o CARP não orientado, Pearn et al. [78], transformaram o CARP com m arestas de procura num VRP com $3m+1$ vértices. Este aumento das dimensões do problema faz com que seja essencialmente teórico o interesse desta transformação. Pearn et al. [78] referiram que este aumento no número de nodos ajuda a compreender a dificuldade acrescida de resolução de um CARP quando comparada com o seu “semelhante” VRP. Por outro lado, pensa-se ser vantajoso ter em atenção as características de cada problema específico, quando do seu estudo, ao invés de considerar casos muito gerais. Assim, nesta tese optou-se por estudar o PRRS como sendo um CARP com restrições adicionais e não tentar adaptar os métodos existentes para o VRP na sua “resolução”.

Tanto quanto se sabe, algoritmos para determinar a solução óptima do CARP não orientado surgiram apenas na década de 90 com os trabalhos de um grupo de Japoneses, Hirabayashi et al. [51] e Saruwatari et al. [85]. Contudo, estes autores só resolvem problemas com um máximo de 15 arestas de procura.

2.4.2. Métodos Heurísticos

No ano de 1973, Christofides [18] apresentou uma heurística de emparelhamentos – *Construct-Strike* – para o CARP, em grafos não orientados. Trata-se, neste caso, de considerar que os carteiros (ou veículos) afectos à distribuição de correio, têm um tempo limite para efectuar o serviço, não podendo, em cada dia, distribuir um número ilimitado de correspondência. Em 1989 Pearn [75] obtém, a partir deste método, um mais eficiente – *Modified Construct-Strike*.

Golden e Wong [42], para além de descrever o CARP, em redes não orientadas, apresentaram também um método heurístico – *Augment-Merge*. Golden et al. [43], [44] apresentaram este algoritmo mais detalhadamente e descreveram um novo – *Path-Scanning*. Pearn [75] apresentou também uma modificação do algoritmo descrito em [44] – *Random Path-Scanning*.

Um novo método heurístico para o CARP, em redes não orientadas, baseado em técnicas de inserção, foi proposto em 1984 por Chapleau et al. [17] – *Parallel-Insert*.

Em 1991, Pearn [76] desenvolveu novos métodos, tendo como objectivo a obtenção de melhores resultados em grafos não orientados, esparsos (de densidade não superior a 30%) e com procuras “elevadas” uma vez que, como afirma, nenhum dos métodos existentes se comporta bem neste tipo de grafos. Assim, apresentou duas versões de uma heurística – *Augment-Insert*. A

heur stica baseia-se no *Augment-Merge Algorithm* de Golden et al [44] e no *Parallel-Insert Algorithm* de Chapleau et al [17]. Neste artigo Pearn comparou as heur sticas: *Construct-Strike*, *Path-Scanning*, *Augment-Merge*, *Modified Construct-Strike*, *Random Path-Scanning* e as duas novas vers es do *Augment-Insert* em funç o de resultados computacionais.

Em 1985 Ulusoy [90] apresentou um algoritmo para o CARP, em que as subrotas admiss veis s o geradas por partiç o de um ciclo "gigante". Esta heur stica, apesar de desenvolvida para o caso n o orientado  , segundo Ulusoy [90], facilmente generaliz vel ao caso orientado.

Mais recentemente, foram apresentadas novas heur sticas para o CARP, em redes n o orientadas, por Benavent et al. [7] – *Cycle-Assignment*; por Bodin e Levy [13], [56] e por Greistorfer [47]. Algoritmos de pesquisa local foram tamb m propostos por Greistorfer [47] e num trabalho conjunto de Eglese e de Li [32], [58].

Em 1994, Greistorfer [48] desenvolveu ainda um algoritmo para o CCPP misto. Lin e Zhao [61], em 1988, fizeram a extens o do algoritmo que apresentam para o CPP orientado ao k -CPP (k carteiros) orientado.

2.4.3. *Minorantes*

Minorantes para problemas com procura nos arcos e restriç es de capacidade podem ser calculados em funç o do valor  ptimo de problemas semelhantes sem restriç es de capacidade. Assim, o valor  ptimo de um CPP   um minorante v lido para o seu hom logo CCPP, podendo um minorante para um CARP ser calculado pela resoluç o de um ARP, quando poss vel.

Mostra-se ent o natural que o c culo de minorantes para CCPP n o orientado seja feito com base na resoluç o de problemas de emparelhamentos perfeitos de custo m nimo, problemas estes que, no geral, fazem parte dos algoritmos existentes para determinaç o da soluç o  ptima do CPP n o orientado [28]. O CPP orientado resolve-se com base em algoritmos de fluxos [28] ou de transporte [6]. Assim, a determinaç o de minorantes para o CCPP e para o CARP pode ser feita tamb m   custa de modelos deste tipo.

Na literatura dispon vel, os poucos artigos em que s o apresentados m todos para determinar minorantes para o CARP focam apenas o caso n o orientado.

O primeiro minorante conhecido foi apresentado por Golden e Wong [42], em 1981, e designado por MLB – *Matching Lower Bound*. Como o pr prio nome indica, o MLB baseia-se na determinaç o de um emparelhamento perfeito de

custo mínimo num grafo, não orientado, induzido pelos vértices de grau ímpar e réplicas do depósito, em número suficiente de forma a garantir que o número de arestas incidentes no depósito é o necessário à criação dos ciclos.

Em 1987 Assad et al. [3] apresentaram o NSLB – *Node Scanning Lower Bound* – para determinar minorantes para o CCPP, também em redes não orientadas. Neste método os vértices são ordenados de acordo com a sua distância ao depósito, sendo cada vértice considerado tantas vezes quanto o seu grau. Assad et al. [3] e Pearn [74] identificaram casos em que o valor obtido com NSLB é igual ao valor óptimo do CARP correspondente.

Um novo minorante baseado numa junção destes dois, e designado por MNSLB⁽¹⁾, foi proposto por Pearn [74]. Win [92] propôs dois novos minorantes, também por sugestão dos dois primeiros. Identificando uns conjuntos de arestas especiais, que denominou por “conjuntos de arestas de corte”, Zaw Win [92] alterou, por um lado o MLB e, por outro, o NSLB e obteve dois novos minorantes, usualmente designados por Zaw1 e Zaw2.

À custa de uma rede expandida em que se consideram réplicas dos vértices e do depósito, ainda focando o caso não orientado, Saruwatari et al. [84] propuseram o NDLB – *Node Duplication Lower Bound*. Este método conduz a um aumento significativo nas dimensões dos problemas, embora seja possível provar que domina o MNSLB, ou seja, o NDLB origina sempre minorantes nunca inferiores aos obtidos com MNSLB.

Benavent et al. [8] desenvolveram três novos métodos, que designaram por LB1, LB2 e LB3. Ao comparar estes métodos com os anteriores provaram que LB1 domina MLB, NSLB e MNSLB; LB2 domina LB1 e Zaw2; e que LB3 domina LB1. Porém a resolução de LB3 passa pela construção de uma rede expandida, semelhante à de NDLB, ou seja, conduz a um aumento bastante significativo nas dimensões dos problemas. Assim, dos métodos desenvolvidos o LB2 é o aconselhado. Neste método LB1 é alterado de forma semelhante à utilizada por Zaw Win [92] para obter Zaw2 a partir de MLB. Por sua vez, LB1 consiste em aplicar as ideias subjacentes a MLB, partindo de uma rede diferente.

Pearn [77] provou que o LB1, proposto por Benavent et al. [8], pode ser encarado como um método exacto para o k-CPP em redes não orientadas, provando assim tratar-se de instâncias do problema resolúveis em tempo polinomial.

(1) Assad e Golden [2] designaram este método por MPLB - *Matching Path Lower Bound*.

No mesmo artigo, Benavent et al. [8] apresentaram ainda um m todo para determinar minorantes, LB4, incorporando as restriç es de capacidade dos ve culos, mas relaxando as que indicam que cada aresta de procura   servida uma e uma s  vez. Este, por m, proporciona, em geral, piores resultados computacionais com maior esforço computacional, dado recorrer a t cnicas de programaç o din mica.

Li e Eglese [57] estudaram um CARP que, embora n o orientado, difere do usual dado o tipo de restriç es que consideram. Enquanto os restantes autores apenas impuseram restriç es de capacidade nos ve culos, neste artigo foram tamb m consideradas restriç es de tempo (relativas ao instante em que cada aresta de "procura" deve ser servida).

2.5. *Outros Problemas*

Orloff [70] relacionando diversos problemas com procuras nos nodos e/ou arestas, definiu, ainda, o problema geral de rotas, GRP – *General Routing Problem*. No GRP pretende determinar-se uma rota de custo m nimo que inclua um determinado conjunto de v rtices e um determinado conjunto de arestas.

Orloff [70] tamb m formalizou e apresentou um algoritmo de partiç es e avaliaç es sucessivas para o GRP. Tratando-se do caso n o orientado o algoritmo exige a resoluç o de um problema de emparelhamento, enquanto o caso orientado exige a resoluç o de um problema de transporte. Male et al. [63] modificaram este algoritmo de forma a ser poss vel garantir a optimalidade da soluç o.

Ainda em 1974, Orloff [71] apresentou o GRP m ltiplo, ou seja, para uma frota com dois ou mais ve culos e mostrou que este problema pode ser transformado num GRP simples equivalente. Esta transformaç o foi, mais tarde, corrigida por Lenstra e Rinnooy Kan [53].

Frederickson et al. [41] definiram um k-CPP diferente no qual, em vez da minimizaç o da dist ncia total das k rotas, se pretende minimizar o comprimento da maior rota. Designaram-no por *minimax* k-CPP.

Jansen [52], em 1993, apresentou heur sticas para o GCRP – *General Capacitated Routing Problem* – ou seja o GRP com restriç es de capacidade. Nesta refer ncia   estabelecida uma an lise de pior caso e s o caracterizados os problemas em que, com as heur sticas apresentadas, a raz o entre o valor  ptimo e o valor da heur stica depende da capacidade dos ve culos, podendo, portanto,

ser calculada antecipadamente, ou seja, antes da resolução dos problemas. Jansen apresentou ainda três métodos heurísticos, assumindo procuras unitárias.

Existem ainda problemas que diferindo dos definidos anteriormente aparecem na literatura com nomes específicos. É disto exemplo o *Stacker Crane Problem* (SCP), definido em 1978 por Frederickson et al. [41], que pode ser visto como um ARP numa rede mista em que todos os arcos são de procura, podendo as arestas ser percorridas em vazio.

O *Windy Postman Problem* (WPP) foi introduzido em 1979 por Minieka [66]. No WPP o custo associado a cada arco depende do sentido deste, sendo referido que o arco ou “está a favor ou contra o vento”.

É também usual surgirem problemas em que é imposta uma relação de precedência no serviço dos arcos de procura, ou uma certa hierarquia entre o serviço de arcos de diferentes subconjuntos do conjunto inicial de arcos de procura. Dror et al. [27] fizeram um estudo detalhado deste tipo de problemas. O limite ao número de “más viragens”, ou mesmo a sua minimização, surge em algumas aplicações práticas deste tipo de problemas (ver por exemplo [11], [12], [65], [82], [86]).

Tentativas de classificação dos problemas definidos e dos algoritmos propostos começaram a surgir na década de setenta. Em 1975, Bodin [9] fez a primeira tentativa de classificação dos problemas tendo em conta o tipo de procura (nos vértices ou nos arcos), o número de veículos e o tipo de abordagem pretendida (algoritmos exactos ou heurísticos). Orloff [70] já tinha estabelecido, em 1974, uma certa ligação entre o TSP, o CPP, o ARP e o GRP.

Em 1981, Bodin e Golden [10] fizeram uma classificação detalhada para problemas de rotas e escalonamento de veículos (VRSP – *Vehicle Routing and Scheduling Problem*). Dividiram ainda em sete classes distintas os métodos existentes. No capítulo dedicado a abordagens heurísticas será adoptada esta classificação para as heurísticas desenvolvidas, que tem sido, desde então, utilizada mesmo em métodos propostos para problemas não necessariamente incluídos nos de VRSP. Desrochers et al. [26], também para o VRSP, desenvolveram sistemas com o objectivo de ajudarem a construir modelos e a decidir qual o algoritmo a usar para os modelos resultantes. Foi dado, neste campo, o primeiro passo para que perante um caso real possa ser escolhido, de forma automatizada, o modelo mais adequado bem como o método de resolução.

No respeitante à classificação dos problemas com procuras nos arcos, Assad e Golden [2] apresentaram uma tabela classificativa bastante detalhada,

considerando todas as combinações de redes possíveis. Referiram ainda diferentes abordagens, quer em termos de determinação de soluções admissíveis, quer em termos de cálculo de minorantes e determinação de soluções óptimas.

2.6. *Aplicações a Situações Reais*

O CARP é um problema de muita aplicabilidade prática. Parece assim natural que mesmo antes de ser definido, por Golden e Wong [42] em 1981, se encontrem na literatura artigos que podem hoje ser encarados como aplicações reais do CARP. Saliente-se, a título ilustrativo, os trabalhos publicados na década de 70 por Beltrami e Bodin [6]; Bodin e Kursh [11], [12]; Male e Liebman [62]; Marks e Liebman [64]; Stern e Dror [87].

Como decerto é compreensível, o maior investimento em problemas com procuras nos arcos encontra-se em abordagens heurísticas. De facto, é recorrendo a estes métodos que se tentam determinar “boas” soluções admissíveis para problemas práticos tais como a leitura dos contadores domésticos de gás, água ou electricidade ([87], [91]); a limpeza automática de ruas, incluindo a lavagem, o tirar a neve, ou a varredura mecânica ([6], [11], [12], [33], [81]); espalhar sal nas ruas nos meses de inverno ([24], [30], [31], [32], [58]); as rotas para as carrinhas de uma escola quando vão buscar os alunos a casa ([17], [68]); a distribuição de correio ao domicílio nos EUA (Bodin e Levy [13], [55] e [56]) e no Canadá (Roy e Rousseau [82] ⁽¹⁾); a recolha de resíduos sólidos ([1], [6], [14], [15], [46], [62], [69], [83], [86]).

São inúmeros os problemas práticos relacionados com a recolha de resíduos sólidos urbanos. Os sistemas de recolha de resíduos sólidos urbanos são “grandes” e complexos, não permitindo grandes experimentações sem um aumento significativo nos custos. Marks e Liebman [64] identificaram muitos destes problemas, apresentando um número significativo de referências bibliográficas. A título ilustrativo consideram-se de seguida alguns desses problemas.

- Determinação do número e tipo de veículos da frota (homogénea ou não). Ulusoy [90] aborda este problema, impondo custos fixos associados à utilização dos diferentes veículos.

⁽¹⁾ Problema designado por CPC – *Canadian Post Corporation*.

- Fixação do número de trabalhadores por veículo, tendo em conta as épocas em que existe maior quantidade de resíduos (como por exemplo no Natal e no Ano Novo) ou menor (Verão nas grandes cidades). Também quando a recolha é feita de segunda a sábado, na segunda feira é usual considerar mais um trabalhador por veículo, pois são resíduos acumulados de dois dias.
- Escolha de uma frequência de recolha. Em Lisboa existem zonas onde a recolha é feita três vezes por semana e outras em que a recolha é feita seis dias por semana.
- Afectação dos pontos de recolha a dias da semana, quando a recolha não é diária ([83]).
- Afectação das rotas de recolha a veículos.
- Localização e número das estações de tratamento ([64]).
- Inclusão ou não de estações intermédias. Após recolhidos, os resíduos são transportados para estas estações intermédias, mais próximas dos locais de recolha. O transporte destas estações para as estações de tratamento é feito por outro tipo de veículos, tendo subjacente outro tipo de abordagem ([46], [64]).
- Afectação das diferentes zonas de recolha a uma ETRS ou a uma das estações intermédias de recolha, caso existam.
- Quando se consideram vários destes problemas, surgem modelos com múltiplos objectivos. Por exemplo, Caruso et al. [16] estudaram a parte do sistema de recolha posterior à recolha no quarteirão considerando múltiplos objectivos, pois tentaram abordar, em simultâneo, as localizações das estações intermédias e finais, o transporte da zona de recolha para as referidas estações intermédias (quando existem) e/ou finais, bem como tecnologias associadas (incineradoras; reciclagem; etc.).
- Uma outra abordagem para o estudo do transporte dos resíduos até à ETRS ou, eventualmente, até às estações intermédias, incluindo tempo de espera na ETRS para despejar os resíduos, é por vezes feita por modelos de simulação ([21], [22]). Fundamentalmente, estes modelos de simulação ajudam a planear o funcionamento de sistemas existentes.
- Na implementação prática dos sistemas há que ter em atenção outro tipo de considerações. A título ilustrativo refira-se:
 - ◊ a minimização do número de más viragens, como por exemplo virar à esquerda ou a inversão de marcha ([11], [12], [65], [82], [86]);
 - ◊ minimização dos “cruzamentos” de veículos, ou seja tentar repartir as zonas de recolha em zonas individuais para os veículos de modo a que estes não se cruzem durante a recolha. Em geral, não convém

que os veículos se encontrem em ruas pequenas dadas as complicações que tal opção pode acarretar a nível de trânsito;

- ◇ rotas equilibradas para os diferentes veículos. Levy e Bodin [55] ao planearem a distribuição de correio tentaram obter rotas equilibradas para os diferentes carteiros. Bodin e Levy [13] fizeram uma partição do grafo de forma a obter subgrafos equilibrados em termos de tempo de trabalho e que não se intersectem muito;
 - ◇ reivindicações sindicais relativas, por exemplo, a horários de trabalho.
- Desenvolvimento de métodos interactivos que permitam incorporar informação nova. Há que ter em consideração o conhecimento prático das pessoas que habitualmente lidam com este tipo de problemas que, de certa forma com razão, não crêem ser possível fazer melhor. Mostram-se necessários sistemas muito interactivos passíveis de responder, em tempo útil, a alterações.
 - Controlar e reduzir a geração de resíduos ([64]) é um dos problemas ecológicos que podem ser considerados. Outro, também importante, é o da separação dos resíduos. Este tipo de problemas passa sempre por uma necessária e difícil mudança de mentalidades da população em geral.
 - Como eliminar os resíduos, ou seja que tipo de tratamento deve ser dado aos diversos tipos de resíduos.
 - Determinação das rotas minimizando o custo da recolha. Só neste subproblema, há que:
 - ◇ definir se a procura é aleatória ou determinística;
 - ◇ distinguir procura uniforme (como na limpeza da neve das ruas, pois esta cai de uma forma uniforme) da não uniforme (como por exemplo na recolha de resíduos);
 - ◇ decidir se a localização das procuras deve ser considerada nos nodos ([64], [89]) ou nos arcos ([1], [86]). O primeiro caso engloba, por exemplo, a determinação das rotas dos veículos que recolhem os vidrões, os papelões, os resíduos industriais, ou seja, todos os contentores que se possam considerar em locais bem distintos. O segundo tipo engloba a recolha da maior parte dos resíduos domésticos em Lisboa;
 - ◇ considerar a rede não orientada ([62]), orientada, ou mista;
 - ◇ verificar a existência de restrições de capacidade para os veículos (por ex., tempo máximo por rota);
 - ◇ identificar os custos (variáveis ou não).

São estes em geral o tipo de problemas referidos em artigos relacionados com a recolha de resíduos ([14]). Não sendo possível tratar em simultâneo todos estes problemas é vulgar que os estudos se centralizem em alguns destes aspectos.

Embora este trabalho se tenha baseado numa aplicação real teve como objectivo fundamental o aspecto teórico do problema. Assim, a situação real descrita no início deste capítulo é modelizada como um problema de programação linear inteira. Todo o estudo subsequente é baseado nesta formalização, relegando para um possível trabalho futuro a tentativa de uma abordagem mais prática e consequentemente mais aplicável à situação real de Lisboa.

Sistemas específicos para a recolha de resíduos em algumas cidades foram já desenvolvidos. Dado o carácter único de cada cidade quer em termos de rede quer em termos de recolha, do tipo de contentores, frota, etc., cidades diferentes podem originar estudos completamente dispare. A título ilustrativo salientem-se os trabalhos desenvolvidos para as cidades de New York, por Beltrami e Bodin [6]; Oyster Bay, por Bodin et al. [14]; Cleveland, por Clark e Gillean [21]; Knoxville, por Male e Liebman [62]; Baltimore, por Marks e Liebman [64]; Izmir, por Or [69]. Recentemente, Álvarez-Valdés et al. [1] apresentaram um sistema para recolha de resíduos que, dizem, se aplica a áreas residenciais afastadas do centro das cidades.

2.7. O Problema de Recolha de Resíduos Sólidos (PRRS)

Neste ponto apresenta-se uma formalização em programação linear inteira para o PRRS, descrito no capítulo anterior. A descrição do PRRS leva-nos a encará-lo como um problema com procura nos arcos com restrições de capacidade. As suas características específicas, nomeadamente a existência do posto de serviço e a representação numa rede orientada, fazem-no diferir dos problemas existentes na literatura, embora pertencente à classe dos CARP com restrições adicionais. A formalização é precedida por definições e pela principal notação utilizada no decorrer do presente texto.

2.7.1. Definições e Notação

Definição 2.1: Define-se *rota* de um veículo como todo o seu percurso desde que sai do Posto de Serviço até lá voltar para deixar a tripulação.

Uma rota de um veículo inclui, pelo menos, duas *subrotas*: a primeira, do Posto de Serviço à Estação de Tratamento de Resíduos Sólidos (ETRS), designa-se por *subrota inicial*; e a última, da ETRS ao Posto de Serviço é designada por *subrota final* do veículo.

Eventualmente, uma rota de um veículo inclui ainda *subrotas intermédias*, da ETRS à ETRS.

Exemplo 2.1: A rota do veículo, esquematizada na figura 1.1, é composta por três subrotas (ver figura 2.3):

- 1ª: Tem início no Posto de Serviço e termina na ETRS – *subrota inicial*. O veículo percorre os arcos 1, 2, ..., 6;
- 2ª: Começa e termina na ETRS – *subrota intermédia*. São percorridos os arcos 7, 8, ..., 15;
- 3ª: Com início na ETRS e fim no Posto de Serviço – *subrota final*. São percorridos os arcos: 16, 17, ..., 22.

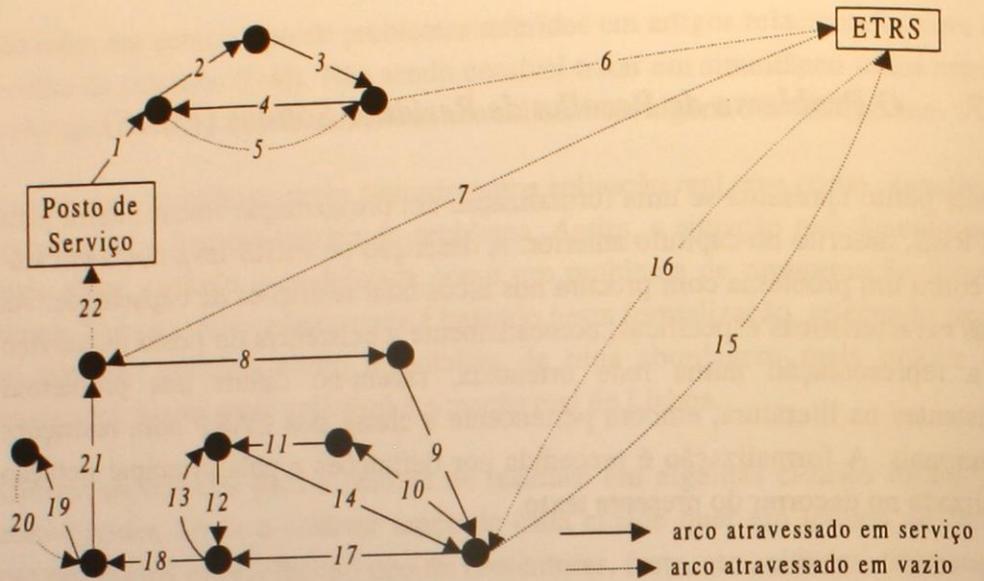


Figura 2.3: Subrotas de uma rota de um veículo

O mapa da zona em estudo, é representado por um grafo orientado, $G = (V, A)$, com $n + 1$ vértices ($|V| = n + 1$), em que:

- o vértice 1 representa o Posto de Serviço;
- o vértice $n + 1$ representa a ETRS;
- os restantes vértices representam cruzamentos ou entroncamentos de ruas ou extremos de ruas sem saída (“becos”). No que se segue, $U = V \setminus \{1, n + 1\}$ representa este conjunto de vértices;
- cada arco, $a = (i, j) \in A$, representa um segmento de rua. O sentido de cada rua é indicado pela orientação do arco, sendo uma rua com dois sentidos representada por dois arcos orientados: (i, j) e (j, i) .

Designa-se por:

P o número total de veículos para efectuar a recolha na zona em estudo (considera-se que p representa um veículo específico, sendo: $p = 1, 2, \dots, P$).

W a capacidade de cada veículo;

$G_R = (V_R, A_R)$, o grafo de procuras, onde:

$A_R \subset A$ é o conjunto dos arcos em que tem de ser efectuada recolha, por vezes designados arcos de “procura”. Considera-se $m = |A_R|$;

$V_R \subset V$ o conjunto dos vértices em que incide pelo menos um arco de A_R ;

$q_{i,j}$ a quantidade de res duos a recolher no arco $(i, j) \in A_R$;

$Q_T = \sum_{(i,j) \in A_R} q_{i,j}$ a quantidade total de res duos a recolher;

$c_{i,j}$ o custo (ou tempo) de recolher res duos no arco $(i, j) \in A_R$;

$C_{A_R} = \sum_{(i,j) \in A_R} c_{i,j}$ o custo de serviço;

$d_{i,j}$ o custo (ou tempo) de passar no arco $(i, j) \in A$ em vazio, ou seja, sem efectuar recolha ($d_{i,j} < c_{i,j}, \forall (i, j) \in A_R$);

$spl(i, j)$ o custo total do caminho de menor custo em vazio de i ($i \in V$) para j ($j \in V$);

$scl(\ell)$ o custo total do circuito de menor custo em vazio que cont m ℓ ($\ell \in V$). Este circuito cont m pelo menos um v rtice $u \neq \ell$.

$d^-(\ell)$ o grau interno do v rtice ℓ , considerando apenas os arcos de procura, ou seja: $d^-(\ell) = |\{(i, \ell): (i, \ell) \in A_R\}|$;

$d^+(\ell)$ o grau externo do v rtice ℓ , considerando apenas os arcos de procura, ou seja: $d^+(\ell) = |\{(\ell, j): (\ell, j) \in A_R\}|$;

$D(\ell) = d^-(\ell) - d^+(\ell)$.

A situaç o real que serviu de base ao presente estudo apresenta as seguintes caracter sticas pr prias:

- A ETRS encontra-se "afastada" da zona onde se pretende programar a recolha de res duos s lidos, n o existindo res duos a recolher nas ruas pr ximas. Logo, $n+1 \notin V_R$, pois todos os arcos incidentes em $n+1$ s o considerados de procura nula, ou seja:

$$d^-(n+1) = d^+(n+1) = 0 \Rightarrow D(n+1) = 0.$$

- Considerando a capacidade limitada dos ve culos, W , e que as rotas terminam no Posto de Serviço, n o incluindo a  ltima passagem de cada ve culo pela ETRS, torna-se  bvio que

$$K = \left\lceil \frac{Q_T}{W} \right\rceil - P \quad (2.1)$$

representa um minorante para o número mínimo de vezes que os P veículos se deslocam à ETRS (exemplo 2.2), sendo $\underline{R} = K + P$ um minorante para o número de subrotas que serão efectuadas, e que se representa por R .

- Note-se ainda que, por hipótese, o número de veículos é baixo, quando comparado com a procura total. Logo, supõe-se que $K + P \geq 2P$, ou seja, cada veículo faz pelo menos a sua subrota inicial e a sua subrota final. Assim, existem sempre P subrotas iniciais e P subrotas finais.

Exemplo 2.2: Considere-se a figura seguinte onde os arcos a cheio representam arcos de procura unitária e os a tracejado arcos de procura nula (que podem, eventualmente ser atravessados em vazio).

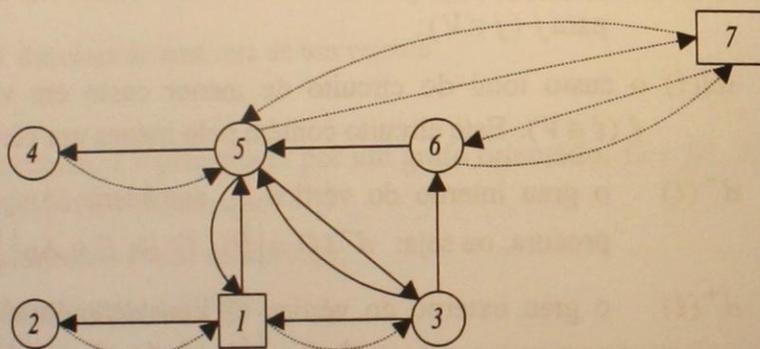


Figura 2.4: Grafo $G = (V, A)$

Suponha-se que existem disponíveis dois veículos de capacidade $W = 2$.

Sendo $Q_T = 9$, são necessárias, pelo menos, $\underline{R} = \left\lceil \frac{Q_T}{W} \right\rceil = \left\lceil \frac{9}{2} \right\rceil = 5$ subrotas.

Como têm que ser usados os dois veículos disponíveis, duas destas 5 subrotas têm que ser finais (de 7 para 1). Logo, os veículos deslocam-se à ETRS três vezes (no fim de cada uma das 2 subrotas iniciais e no fim da subrota intermédia, a ser atribuída a um dos veículos). ♦

Um conjunto de rotas que minimize o custo total de recolha dos resíduos sólidos, pode obter-se pela resolução de um CARP, com restrições adicionais, em $G = (V, A)$. Contudo, a resolução do CARP é particularmente difícil, pois Golden e Wong [42] provaram que até a sua 0.5-*aproximação*, que consiste em determinar uma solução de valor nunca superior a 1.5 do valor óptimo, é um problema NP-difícil.

As características específicas do problema em estudo fazem-no diferir do CARP que usualmente se define [42]. Tal motivou a designação de PRRS referida e que se utilizará no decorrer desta tese.

2.7.2. *Formalização em Programação Linear Inteira*

A aplicação de recolha de resíduos sólidos descrita vai ser formalizada como um problema de programação linear inteira, PRRS. Usualmente, em problemas deste género, definem-se dois tipos de variáveis: variáveis binárias que indicam como é efectuado o serviço; e variáveis inteiras que indicam o número de vezes que cada rua é atravessada em vazio.

Na recolha de resíduos sólidos cada veículo realiza uma rota formada por um certo número de subrotas. Assim, o serviço de cada veículo depende do número de subrotas e de rotas que serão necessárias. Sendo desconhecido o número de subrotas convém, antes de mais, estabelecer intervalos para a sua variação. A definição de variáveis é então precedida da majoração do número de subrotas a efectuar por cada veículo.

Considere-se que os $m = |A_R|$ arcos de procura estão ordenados por ordem não crescente das procuras, isto é, tais que: $q_1 \geq q_2 \geq \dots \geq q_m$.

Seja ℓ o número máximo de arcos compatível com a capacidade de um veículo que iniciando por servir o arco de maior procura associada, sirva os arcos pela ordem por que foram ordenados. Defina-se então, ℓ tal que:

$$\sum_{a=1}^{\ell} q_a \leq W < \sum_{a=1}^{\ell+1} q_a.$$

Nestas condições, torna-se óbvio que $L = \left\lceil \frac{m}{\ell} \right\rceil$ é um majorante para RT , o número total de subrotas em qualquer solução óptima.

Considere-se que \bar{R} representa um majorante para o número de subrotas a efectuar por qualquer dos veículos, em qualquer solução óptima. Assim, se R_p representar o número de subrotas que na solução óptima serão realizadas pelo veículo p ($p = 1, \dots, P$), tem-se $R_p \leq \bar{R}$. Na proposição seguinte estabelece-se um limite para este valor desconhecido.

Proposição 2.1: Se $\bar{R} = L - 2(P - 1) \Rightarrow R_p \leq \bar{R}, \forall p$.

Prova: Por hipótese, todos os veículos são usados, realizando pelo menos duas subrotas, uma inicial e uma final. Por outro lado, nenhuma solução óptima inclui subrotas em vazio. Logo, $P - 1$ dos veículos são afectos a $2(P - 1)$ subrotas, pelo menos. Neste caso, o veículo que falta considerar não realiza mais de $L - 2(P - 1)$ subrotas.

Assim, $L - 2(P - 1)$ representa um majorante para o número de subrotas a efectuar por um qualquer dos veículos, em qualquer solução óptima.

Logo, $\bar{R} = L - 2(P - 1) \Rightarrow R_p \leq \bar{R}, \forall p$. □

Definam-se, então, para cada veículo, $p = 1, \dots, P$, em cada uma das suas possíveis subrotas, $r = 1, \dots, \bar{R}$, as variáveis:

$$\bullet x_{i,j}^{r,p} = \begin{cases} 1 & \text{se o veículo } p \text{ recolhe os resíduos de } (i, j) \text{ durante a subrota } r \\ 0 & \text{c.c.} \end{cases}$$

$$\forall (i, j) \in A_R;$$

$$\bullet y_{i,j}^{r,p} = \text{número de vezes que, durante a subrota } r, \text{ o veículo } p \text{ atravessa o arco } (i, j) \in A \text{ em vazio, ou seja, sem recolher resíduos.}$$

O problema de determinação das rotas a efectuar pelos veículos que recolhem os resíduos na zona em estudo, de forma a minimizar o custo total de recolha, pode então ser formalizado como se segue.

(PRRS)

$$\text{Min } z = \sum_{p=1}^P \sum_{r=1}^{\bar{R}} \sum_{(i,j) \in A_R} c_{i,j} x_{i,j}^{r,p} + \sum_{p=1}^P \sum_{r=1}^{\bar{R}} \sum_{(i,j) \in A} d_{i,j} y_{i,j}^{r,p} \quad (2.2)$$

s.a:

$$\sum_{(i,\ell) \in A_R} x_{i,\ell}^{r,p} + \sum_{(i,\ell) \in A} y_{i,\ell}^{r,p} - \sum_{(\ell,i) \in A_R} x_{\ell,i}^{r,p} - \sum_{(\ell,i) \in A} y_{\ell,i}^{r,p} = 0 \quad \forall i \in U; \quad (2.3)$$

$$p = 1, \dots, P;$$

$$r = 1, \dots, \bar{R};$$

$$\sum_{r=1}^{\bar{R}} \left(\sum_{(i,\ell) \in A_R} x_{i,\ell}^{r,P} + \sum_{(i,\ell) \in A} y_{i,\ell}^{r,P} - \sum_{(\ell,i) \in A_R} x_{\ell,i}^{r,P} - \sum_{(\ell,i) \in A} y_{\ell,i}^{r,P} \right) = 0 \quad (2.4)$$

$$i \in \{1, n+1\}; \\ p = 1, \dots, P;$$

$$\sum_{(1,\ell) \in A_R} x_{1,\ell}^{1,P} + \sum_{(1,\ell) \in A} y_{1,\ell}^{1,P} = \sum_{(\ell,1) \in A_R} x_{\ell,1}^{1,P} + \sum_{(\ell,1) \in A} y_{\ell,1}^{1,P} + 1 \quad p = 1, \dots, P; \quad (2.5)$$

$$\sum_{(\ell, n+1) \in A} y_{\ell, n+1}^{1,P} = 1 \quad p = 1, \dots, P; \quad (2.6)$$

$$\sum_{(n+1, \ell) \in A} y_{n+1, \ell}^{r,P} \leq 1 \quad p = 1, \dots, P; \quad r = 2, \dots, \bar{R}; \quad (2.7)$$

$$\sum_{(n+1, \ell) \in A} y_{n+1, \ell}^{r,P} \geq \sum_{(\ell, n+1) \in A} y_{\ell, n+1}^{r,P} \quad p = 1, \dots, P; \quad r = 2, \dots, \bar{R}; \quad (2.8)$$

$$\sum_{(n+1, \ell) \in A} y_{n+1, \ell}^{r,P} \geq \sum_{(n+1, \ell) \in A} y_{n+1, \ell}^{r+1, P} \quad p = 1, \dots, P; \quad r = 2, \dots, \bar{R} - 1; \quad (2.9)$$

$$\sum_{(\ell, n+1) \in A} y_{\ell, n+1}^{r,P} \geq \sum_{(\ell, n+1) \in A} y_{\ell, n+1}^{r+1, P} \quad p = 1, \dots, P; \quad r = 2, \dots, \bar{R} - 1; \quad (2.10)$$

$$\sum_{i \in S} \sum_{j \in S} (x_{i,j}^{r,P} + y_{i,j}^{r,P}) \leq M \left[\sum_{i \in S} \sum_{j \in S} (x_{i,j}^{r,P} + y_{i,j}^{r,P}) + \sum_{i \notin S} \sum_{j \in S} (x_{i,j}^{r,P} + y_{i,j}^{r,P}) \right] \quad (2.11)$$

$$\forall S \subseteq V \setminus \{n+1\}; \quad p = 1, \dots, P; \quad r = 1, \dots, \bar{R};$$

$$\sum_{(i,j) \in A_R} q_{i,j} x_{i,j}^{r,P} \leq W \quad p = 1, \dots, P; \quad r = 1, \dots, \bar{R}; \quad (2.12)$$

$$\sum_{p=1}^P \sum_{r=1}^{\bar{R}} x_{i,j}^{r,P} = 1 \quad \forall (i,j) \in A_R; \quad (2.13)$$

$$x_{i,j}^{r,P} \in \{0, 1\} \quad \forall (i,j) \in A_R; \quad p = 1, \dots, P; \quad r = 1, \dots, \bar{R}; \quad (2.14)$$

$$y_{i,j}^{r,P} \geq 0 \text{ e inteiros} \quad \forall (i,j) \in A; \quad p = 1, \dots, P; \quad r = 1, \dots, \bar{R}. \quad (2.15)$$

As restrições (2.3) e (2.4) são necessárias para garantir a continuidade das subrotas, em cada vértice do grafo associado.

A primeira subrota de cada veículo é definida como sendo uma subrota do posto de serviço para a ETRS, nas condições (2.5) e (2.6).

Nas restrições (2.7) e (2.8) caracterizam-se as subrotas intermédias e final de cada veículo. São então definidos alguns dos arcos incidentes em $n+1$, tendo em conta que as subrotas intermédias começam e terminam na ETRS, enquanto as finais são da ETRS para o posto de serviço.

As subrotas geradas são ordenadas por ordem crescente dos índices, pelas condições (2.9) e (2.10).

Sendo M uma constante suficientemente elevada, as condições (2.11) evitam a geração de subrotas não admissíveis, ou seja de subrotas que não incluem o vértice $n+1$ (figura 2.5).

Exemplo 2.3: Considere-se um grafo com $n+1=8$ vértices e as seguintes subrotas para um veículo, p :

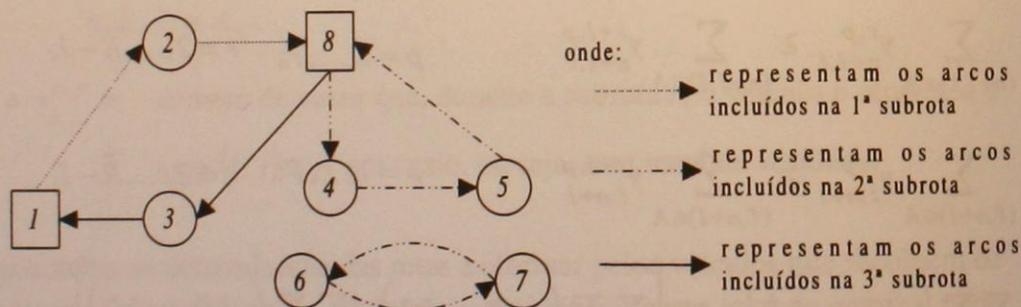


Figura 2.5: Subrotas não admissíveis

Se $r=2$ e $S = \{6, 7\}$, vem:

$$\sum_{i \in S} \sum_{j \in S} (x_{i,j}^{2,p} + y_{i,j}^{2,p}) = 2 > 0$$

$$e \quad \sum_{i \in S} \sum_{j \notin S} (x_{i,j}^{2,p} + y_{i,j}^{2,p}) + \sum_{i \notin S} \sum_{j \in S} (x_{i,j}^{2,p} + y_{i,j}^{2,p}) = 0,$$

sendo violadas as restrições (2.11). ◆

Finalmente, garante-se que a capacidade dos veículos não é excedida, (2.12), e que os resíduos existentes em cada rua são recolhidos exactamente uma vez por um e um só veículo, (2.13).

A minimização do custo total é obviamente garantida com a minimização da função objectivo, (2.2).

Tipicamente as restrições consideradas nas formalizações do CARP [35] garantem:

- o equilíbrio de cada vértice (semelhantes a (2.3) e (2.4)), ou seja, garantem que o grau interno de cada vértice é igual ao seu grau externo, de forma a ser possível gerar circuitos admissíveis.
- a inexistência de subrotas ilegais. No geral este conceito está associado à impossibilidade de existência de subrotas isoladas. No PRRS o conceito de subrota ilegal está relacionado com a inclusão do vértice especial $n + 1$ em cada subrota, conforme referido.
- que a capacidade dos veículos não é excedida.
- e que os arcos de procura são servidos uma única vez.

No PRRS consideraram-se ainda as restrições que definem as subrotas, diferenciando, para cada veículo, a sua subrota inicial. O objectivo, tal como o identificado, consiste, habitualmente, na minimização do custo total.

Capítulo 3. Relaxações para o PRRS

Relaxações de formalizações de problemas como o PRRS proporcionam a determinação de minorantes para o seu valor óptimo. Torna-se assim também possível analisar as soluções admissíveis em função dos seus valores, estabelecendo desvios para os erros relativos. Contudo, quando a procura se localiza nos arcos e são consideradas restrições adicionais, por exemplo de capacidade, o desenvolvimento de métodos que forneçam minorantes para o valor óptimo tem estado votado ao abandono quando, por exemplo, comparado com métodos heurísticos para problemas do mesmo tipo.

3.1. *Introdução*

Minorantes para o valor óptimo de problemas com procura nos arcos e restrições de capacidade podem ser calculados em função do valor óptimo de problemas semelhantes sem restrições de capacidade. Assim, o valor óptimo de um CPP é um minorante válido para o valor óptimo do seu homólogo CCPP, enquanto um minorante para o valor óptimo do CARP pode ser calculado a partir da resolução de um ARP.

Parece então natural que o cálculo de minorantes para os valores óptimos do CCPP e do CARP não orientado seja feito com base na resolução de problemas de emparelhamentos perfeitos de custo mínimo. Pois, em geral, a determinação da solução óptima do CPP não orientado [28] exige a resolução de emparelhamentos perfeitos de custo mínimo. O CPP orientado resolve-se com base em algoritmos de fluxos [28] ou de transporte [6]. Assim, a determinação de minorantes para os valores óptimos do CCPP e do CARP pode também ser feita à custa de modelos deste tipo.

Neste capítulo começa por se deduzir uma relaxação para o problema. A agregação de restrições, na formalização apresentada, e a omissão das condições de conexidade, permite a subdivisão do problema em dois subproblemas. Sendo um deles trivial, o outro consegue-se provar ser equivalente a um problema de transporte.

O algoritmo LB, utilizado para a determinação de minorantes para o valor óptimo do PRRS, que se apresenta e exemplifica no terceiro ponto, baseia-se nesta relaxação. O exemplo aqui criado para a ilustração do algoritmo será também utilizado na exemplificação das heurísticas.

A inclusão de parte das restrições omitidas, de conexidade e de capacidade, na relaxação anterior origina uma nova relaxação, que se apresenta no quarto ponto deste capítulo. Tratando-se ainda de um problema de "difícil" resolução são consideradas novas relaxações. Consideradas duas abordagens diferentes obtêm-se dois novos modelos passíveis de fornecerem minorantes para o valor óptimo do PRRS em tempo polinomial. O primeiro destes modelos pode ser resolvido, tal como a primeira relaxação, por um problema de transporte e o segundo por um problema de fluxo de custo mínimo.

Neste capítulo são feitas algumas comparações entre os valores dos minorantes para o valor óptimo do PRRS que se podem obter com base nas relaxações. Como se verá, em termos teóricos nada se pode garantir da dominância de nenhuma das relaxações.

3.2. Relaxação com Inclusão de Restrições de Equilíbrio e de Capacidade

Substituindo as condições relativas à capacidade de cada veículo em cada subrota por restrições relativas ao conjunto de subrotas de cada veículo, está-se a relaxar o problema (2.2)–(2.15).

Para $p = 1, \dots, P$, seja:

$$\bullet \quad x_{i,j}^p = \sum_{r=1}^{\bar{R}} x_{i,j}^{r,p} = \begin{cases} 1 & \text{se o veículo } p \text{ recolhe os resíduos de } (i, j) \\ 0 & \text{c.c.} \end{cases}$$

$$\forall (i, j) \in A_R;$$

$$\bullet \quad y_{i,j}^p = \sum_{r=1}^{\bar{R}} y_{i,j}^{r,p}$$

representa o número de cópias do arco $(i, j) \in A$ que devem ser consideradas na p -ésima rota, representando o número de vezes que o veículo p atravessa o segmento de rua (i, j) em vazio, i.e., sem efectuar recolha;

- $R_p =$ número de subrotas da p -ésima rota, ou seja, feitas pelo veículo p .
Note-se que $2 \leq R_p \leq \bar{R}$, $\forall p = 1, \dots, P$ e que cada veículo visita a ETRS $R_p - 1$ vezes. Pois, no final da recolha, cada veículo passa primeiro pelo posto de serviço, antes de ir à ETRS pela última vez.

Por outro lado, tendo em conta as restrições (2.7) e (2.9) referentes a todas as subrotas com excepção da inicial, torna-se óbvio que

$$R_p - 1 = \sum_{r=2}^{\bar{R}} \sum_{(n+1, \ell) \in A} y_{n+1, \ell}^{r,p}.$$

Esta definição de variáveis, correspondendo a um agrupamento das subrotas de cada veículo, faz com que se deixe de ter informação sobre cada uma das subrotas, ficando apenas identificáveis as rotas no seu conjunto. Assim, sendo possível identificar os arcos percorridos por cada um dos veículos, não se conseguem individualizar as suas subrotas.

Agregando então as restrições para o conjunto das subrotas de cada rota (veículo) obtém-se:

$$\text{Min } z_R = \sum_{p=1}^P \sum_{(i,j) \in A_R} c_{i,j} x_{i,j}^p + \sum_{p=1}^P \sum_{(i,j) \in A} d_{i,j} y_{i,j}^p \quad (3.1)$$

s.a:

$$\left\{ \begin{array}{l} \sum_{(i,\ell) \in A_R} x_{i,\ell}^p + \sum_{(i,\ell) \in A} y_{i,\ell}^p - \sum_{(\ell,i) \in A_R} x_{\ell,i}^p - \sum_{(\ell,i) \in A} y_{\ell,i}^p = 0 \quad \forall i \in V; \\ p = 1, \dots, P; \end{array} \right. \quad (3.2)$$

$$\left\{ \begin{array}{l} \sum_{(l,\ell) \in A_R} x_{l,\ell}^p + \sum_{(l,\ell) \in A} y_{l,\ell}^p \geq 1 \quad p = 1, \dots, P; \end{array} \right. \quad (3.3)$$

$$\left\{ \begin{array}{l} \sum_{(n+1,\ell) \in A} y_{n+1,\ell}^p = R_p - 1 \quad p = 1, \dots, P; \end{array} \right. \quad (3.4)$$

$$\left\{ \begin{array}{l} \sum_{i \in S} \sum_{j \in S} (x_{i,j}^p + y_{i,j}^p) \leq M \sum_{i \in S} \sum_{j \notin S} (x_{i,j}^p + y_{i,j}^p) \quad \forall S \subseteq V \setminus \{n+1\}; \\ p = 1, \dots, P; \end{array} \right. \quad (3.5)$$

$$\left\{ \begin{array}{l} \sum_{(i,j) \in A_R} q_{i,j} x_{i,j}^p \leq R_p W \quad p = 1, \dots, P; \end{array} \right. \quad (3.6)$$

$$\left\{ \begin{array}{l} \sum_{p=1}^P x_{i,j}^p = 1 \quad \forall (i,j) \in A_R; \end{array} \right. \quad (3.7)$$

$$\left\{ \begin{array}{l} x_{i,j}^p \in \{0, 1\} \quad \forall (i,j) \in A_R; p = 1, \dots, P; \end{array} \right. \quad (3.8)$$

$$\left\{ \begin{array}{l} y_{i,j}^p \geq 0 \text{ e inteiros} \quad \forall (i,j) \in A; p = 1, \dots, P; \end{array} \right. \quad (3.9)$$

$$\left\{ \begin{array}{l} R_p \geq 2 \text{ e inteiros} \quad p = 1, \dots, P. \end{array} \right. \quad (3.10)$$

Note-se que o valor óptimo deste problema, dada a forma como foi construído, representa um minorante para o valor óptimo do problema inicial, PRRS, ou seja: $z_R^* \leq z^*$.

Considere-se agora o problema resultante do anterior por omissão das restrições (3.5) e agregação das restantes para o conjunto das P rotas:

$$\sum_{p=1}^P \left(\sum_{(i,\ell) \in A_R} x_{i,\ell}^p + \sum_{(i,\ell) \in A} y_{i,\ell}^p - \sum_{(\ell,i) \in A_R} x_{\ell,i}^p - \sum_{(\ell,i) \in A} y_{\ell,i}^p \right) = 0 \quad \forall i \in V \quad (3.2')$$

$$\sum_{p=1}^P \sum_{(l,\ell) \in A_R} x_{l,\ell}^p + \sum_{p=1}^P \sum_{(l,\ell) \in A} y_{l,\ell}^p \geq P \quad (3.3')$$

$$\sum_{p=1}^P \sum_{(n+1,\ell) \in A} y_{n+1,\ell}^p = \sum_{p=1}^P R_p - P \quad (3.4')$$

$$\sum_{p=1}^P \sum_{(i,j) \in A_R} q_{i,j} x_{i,j}^p \leq W \sum_{p=1}^P R_p \quad (3.6')$$

$$\sum_{p=1}^P x_{i,j}^p = 1 \quad \forall (i,j) \in A_R; \quad (3.7)$$

$$x_{i,j}^p \in \{0,1\} \quad \forall (i,j) \in A_R; \quad p = 1, \dots, P; \quad (3.8)$$

$$y_{i,j}^p \geq 0 \text{ e inteiros} \quad \forall (i,j) \in A; \quad p = 1, \dots, P; \quad (3.9)$$

$$R_p \geq 2 \text{ e inteiros} \quad p = 1, \dots, P. \quad (3.10)$$

Como $\sum_{p=1}^P \sum_{(i,j) \in A_R} q_{i,j} x_{i,j}^p = Q_T$, (3.6') pode ser escrita na forma:

$$\sum_{p=1}^P R_p \geq \left\lceil \frac{Q_T}{W} \right\rceil.$$

Em consequência, o valor óptimo de (3.1) continua a representar um minorante válido para o valor óptimo de PRRS se (3.4') for substituída por:

$$(3.4'') \quad \sum_{p=1}^P \sum_{(n+1,\ell) \in A} y_{n+1,\ell}^p \geq \left\lceil \frac{Q_T}{W} \right\rceil - P = K.$$

Por outro lado, em (3.2') e (3.3'), $\sum_{p=1}^P \sum_{(i,\ell) \in A_R} x_{i,\ell}^p$ representa o número de arcos de A_R com vértice inicial i . Logo

$$\sum_{p=1}^P \sum_{(i,\ell) \in A_R} x_{i,\ell}^p = d^+(i) \quad , \quad \sum_{p=1}^P \sum_{(\ell,i) \in A_R} x_{\ell,i}^p = d^-(i)$$

e o conjunto de restrições anterior pode ainda simplificar-se.

Com estas transformações as variáveis $x_{i,j}^p$, estando isoladas num dos termos da função objectivo (3.1), ficam também isoladas nas restrições (3.7) e (3.8). O mesmo se verifica no respeitante às restantes variáveis $y_{i,j}^p$ que estão isoladas tanto na função objectivo como nas restrições (3.2'), (3.3'), (3.4'') e (3.9). O problema anterior pode então subdividir-se nos dois subproblemas que se apresentam de seguida.

• Subproblema (A):

$$\text{Min } z_A = \sum_{p=1}^P \sum_{(i,j) \in A_R} c_{i,j} x_{i,j}^p \quad (3.1A)$$

s.a:

$$\left\{ \begin{array}{l} \sum_{p=1}^P x_{i,j}^p = 1 \quad \forall (i,j) \in A_R; \end{array} \right. \quad (3.7)$$

$$\left\{ \begin{array}{l} x_{i,j}^p \in \{0,1\} \quad \forall (i,j) \in A_R; p = 1, \dots, P. \end{array} \right. \quad (3.8)$$

Resolver este subproblema não é mais que determinar o custo total dos arcos de procura, ou seja que calcular C_{A_R} .

• Subproblema (B):

$$\text{Min } z_B = \sum_{p=1}^P \sum_{(i,j) \in A} d_{i,j} y_{i,j}^p \quad (3.1B)$$

s.a :

$$\left\{ \sum_{p=1}^P \left(\sum_{(i,\ell) \in A} y_{i,\ell}^p - \sum_{(\ell,i) \in A} y_{\ell,i}^p \right) = d^-(i) - d^+(i) \equiv D(i) \quad \forall i \in V \quad (3.2')$$

$$\left\{ \sum_{p=1}^P \sum_{(I,\ell) \in A} y_{I,\ell}^p \geq P - d^+(I) \quad (3.3')$$

$$\left\{ \sum_{p=1}^P \sum_{(n+1,\ell) \in A} y_{n+1,\ell}^p \geq K \quad (3.4'')$$

$$\left\{ y_{i,j}^p \geq 0 \quad \text{e inteiros} \quad \forall (i,j) \in A; p = 1, \dots, P. \quad (3.9)$$

Este subproblema corresponde a identificar o número de vezes que cada arco deve ser copiado (ou seja, que cada segmento de rua deve ser percorrido em vazio), garantindo:

- o equilíbrio de todos os vértices definido em (3.2'). Entende-se por *vértice equilibrado* (ou *simétrico*) um vértice em que os graus interno e externo são iguais, tendo em conta tanto os arcos de procura como as cópias dos arcos;
- a existência de pelo menos P arcos com vértice inicial I , (3.3');
- a existência de pelo menos K arcos com vértice inicial $n+1$, (3.4'');

minimizando o custo associado aos arcos copiados (ou seja, o custo do percurso em vazio), (3.1B).

Note-se que a solução obtida com a resolução destes dois subproblemas pode ser não admissível para o problema inicial, PRRS. Não se consegue, por exemplo, garantir a existência de P subrotas iniciais nem de P subrotas finais. Embora com a resolução de (B) o vértice I fique com grau (quer interno, quer externo) mínimo igual a P , nada garante que se possa criar uma subrota de I para $n+1$, pois estes vértices podem não pertencer à mesma componente conexa, no multigrafo solução. Por outro lado, o simples facto de impor o equilíbrio de todos os vértices, não garante que estes não fiquem isolados, não sendo possível incorporá-los em nenhuma subrota admissível (intermédia, inicial ou final).

Exemplo 3.1: Suponha-se disponível um veículo ($P = 1$) de capacidade $W = 3$.

Considerem-se todas as procuras unitárias, ou seja $q_a = 1, \forall a \in A_R$.

Se, na figura 3.1, os arcos a cheio forem os de procura, vem $Q_T = 5$.

Logo, o número mínimo de subrotas é $\underline{R} = \left\lceil \frac{5}{3} \right\rceil = 2$.

Considere-se que os arcos a tracejado da figura seguinte representam os arcos resultantes da resolução do subproblema (B).

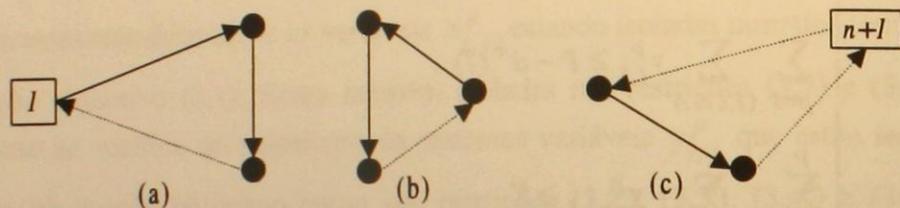


Figura 3.1: Multigrafo associado à resolução de (B)

Neste caso não é possível gerar nem a subrota inicial (de l para $n+1$), nem a subrota final do veículo (de $n+1$ para l), dado os vértices l e $n+1$ não pertencerem à mesma componente conexa.

Por outro lado, a componente isolada (b) também não representa uma subrota admissível para o veículo, pois não inclui a ETRS (o vértice $n+1$). ♦

A resolução de (B) também não obriga a que sejam satisfeitas as restrições de capacidade dos veículos em cada subrota. Tendo-se perdido informação sobre as subrotas, nada garante ser possível gerar, a partir da solução de (B), subrotas em que a soma das procuras dos arcos nelas incluídos não exceda W .

Apenas se pode afirmar que o multigrafo associado à solução de (B), conjuntamente com os arcos de procura, é *equilibrado* ou *simétrico* (ou seja, todos os vértices têm grau interno igual ao externo), sendo os graus dos vértices l e $n+1$ nunca inferiores a um certo valor. O grau mínimo para o vértice l é imposto na tentativa de ter pelo menos o grau mínimo necessário para a geração das P subrotas iniciais e das P subrotas finais. O grau imposto para o vértice $n+1$ está relacionado com o número mínimo de subrotas que se sabem existir. Contudo, nada garante que seja possível a geração de subrotas admissíveis, nem das P rotas para os P veículos, neste multigrafo.

Assim, a solução da relaxação apresentada representando, em geral, uma solução não admissível para o PRRS, tem valor nunca superior ao valor óptimo do PRRS. Ou seja, a soma dos valores óptimos dos dois subproblemas representa um minorante para o valor óptimo do PRRS. Tendo (A) solução trivial, vai de seguida desenvolver-se um modelo para a resolução de (B).

Represente-se por $f_{i,j}$ o número de cópias do arco $(i, j) \in A$ nas P rotas. Então

$$f_{i,j} = \sum_{p=1}^P y_{i,j}^p,$$

e o subproblema anterior pode ser rescrito na forma:

$$(B) \quad \text{Min } z_B = \sum_{(i,j) \in A} d_{i,j} f_{i,j} \quad (3.1B)$$

s.a :

$$\left\{ \begin{array}{l} \sum_{(i,\ell) \in A} f_{i,\ell} - \sum_{(\ell,i) \in A} f_{\ell,i} = D(i) \quad \forall i \in V \quad (3.2') \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{(I,\ell) \in A} f_{I,\ell} \geq P - d^+(I) \quad (3.3') \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{(n+1,\ell) \in A} f_{n+1,\ell} \geq K \quad (3.4'') \end{array} \right.$$

$$\left\{ \begin{array}{l} f_{i,j} \geq 0 \text{ e inteiros} \quad \forall (i, j) \in A. \quad (3.9') \end{array} \right.$$

Está-se perante um problema que tem as restrições habituais dos problemas de fluxos (lei de Kirchhoff), (3.2'), e duas restrições adicionais: (3.3') e (3.4'').

Pretende construir-se um novo problema (B') equivalente a (B) (de acordo com a definição que se apresenta de seguida) mas que apenas contenha restrições do tipo (3.2'). Assim, torna-se simples resolver (B'), podendo este ainda ser transformado num problema de transporte equivalente, como se verá.

Definição 3.1: Dois *problemas* dizem-se *equivalentes* se tiverem o mesmo valor óptimo e a cada solução admissível de um deles corresponder uma solução admissível do outro.

De forma análoga ao que Beltrami e Bodin [6] consideraram, em (B) os vértices de U ($U = V \setminus \{I, n+1\}$) representam origens se forem vértices

desequilibrados por terem, no grafo G_R , grau interno superior ao grau externo. Representam destinos se tiverem grau externo superior ao interno em G_R .

A partir de (3.2') e (3.4'') e relembrando que $D(n+1) = 0$, pode então considerar-se que $n+1$ representa, simultaneamente, uma origem e um destino de (B) com oferta e procura K .

O vértice I pode ser origem e/ou destino, consoante a instância particular em estudo. Os casos que se podem verificar para este vértice são identificados em anexo (A1), sendo resumidos na tabela 3.1.

Se		I é	oferta / procura ($a(I) / b(I)$)
$D(I) = 0$	$P > d^+(I) = d^-(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$
$D(I) > 0$	$P \leq d^-(I)$	origem	$a(I) = D(I)$
	$P > d^-(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$
$D(I) < 0$	$P \leq d^+(I)$	destino	$b(I) = -D(I)$
	$P > d^+(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$

Tabela 3.1: Casos que se podem verificar para o vértice I

De forma a simplificar a exposição, no que se segue supõe-se estar perante uma rede em que:

$$D(I) = d^-(I) - d^+(I) > 0 \quad \text{e} \quad d^-(I) < P.$$

Logo, sem perda de generalidade, e salvo indicação contrária, I representará uma origem de (B) com oferta $a(I) = P - d^+(I)$ e um destino com procura $b(I) = P - d^-(I)$.

As provas e deduções que se fazem para este caso, adaptam-se bem a qualquer dos casos restantes. A validade do minorante foi provada, para qualquer dos casos possíveis, por Mourão e Almeida [67].

Uma forma de construir um problema de simples resolução (B') , a partir do subproblema (B) , consiste em desdobrar cada um dos vértices que represente simultaneamente uma origem e um destino em dois, um representando esse vértice como origem e o outro representando-o como destino.

Genericamente, a construção de (B') envolve:

- Criar um novo vértice, $(n+1)'$;
 Criar um arco, de custo nulo, de $n+1$ para $(n+1)'$;
 Substituir cada arco $(n+1, \ell) \in A$ por um arco $((n+1)', \ell)$ com igual custo;

Assim, o vértice $(n+1)'$ representa a origem $n+1$, com oferta K , e o vértice $n+1$ representa o destino $n+1$, com procura K .

- Criar um novo vértice, l' ;
 Criar um arco, de custo nulo, de l para l' ;
 Substituir cada arco $(l, \ell) \in A$ por um arco (l', ℓ) com o mesmo custo;
- Assim, o vértice l' representa a origem l , com oferta $P - d^+(l)$, e o vértice l representa o destino l , com procura $P - d^-(l)$.

Exemplo 3.2: Considere-se um grafo onde os arcos que podem ser percorridos em vazio, e os custos respectivos, são:

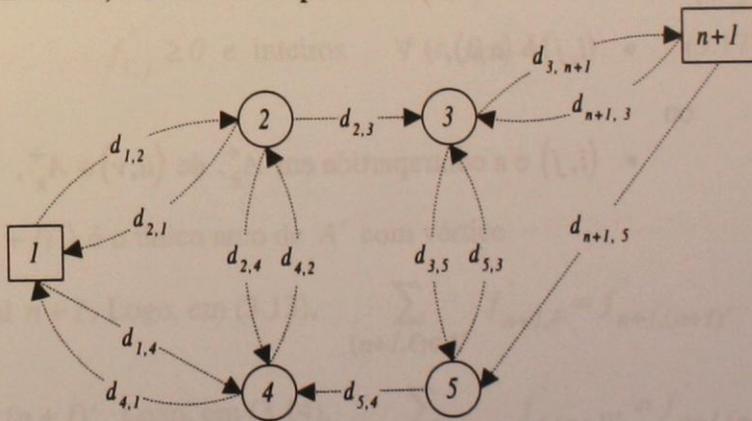


Figura 3.2: Arcos que podem ser percorridos em vazio

Para construir (B') há que transformar a rede anterior em:

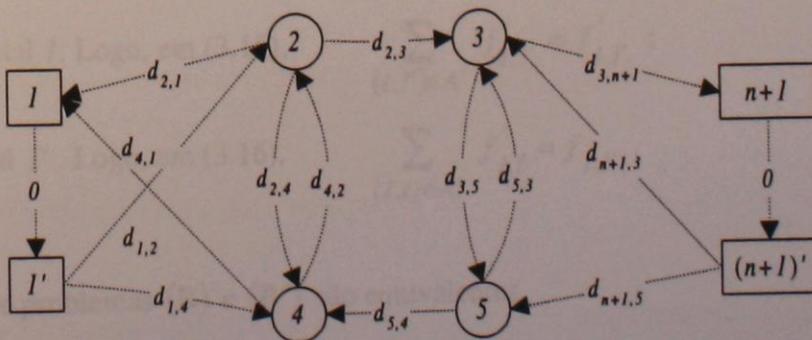


Figura 3.3: Rede transformada para passar de (B) a (B')

Considere-se então o grafo $G' = (V', A')$ definido por:

- $V' = V \cup \{I', (n+1)'\}$;
- $A' = \left[A \setminus (A_{n+1}^+ \cup A_I^+) \right] \cup \{(n+1, (n+1)')\} \cup \{(I, I')\} \cup A_{(n+1)'}^+ \cup A_{I'}^+$

onde, para $k \in \{I, n+1\}$, $A_k^+ = \{(k, \ell) \in A : \ell \in V\}$

e $A_{k'}^+ = \{(k', \ell) : (k, \ell) \in A \wedge \ell \in V\}$.

Sejam $d'_{i,j}$ os custos unitários associados aos arcos de A' , tais que:

$$d'_{i,j} = \begin{cases} d_{u,v} & \text{se } (i, j) \in A' \text{ resulta de } (u, v) \in A \\ 0 & \text{c.c.} \end{cases}$$

onde $(i, j) \in A'$ resulta de $(u, v) \in A$ se:

- $(i, j) = (u, v)$

ou

- (i, j) é a contrapartida em A_k^+ , de $(u, v) \in A_k^+$, $k \in \{I, n+1\}$.

Como se prova no lema seguinte resolver o problema (B) equivale a resolver:

$$(B') \quad \text{Min } z_{B'} = \sum_{(i,j) \in A'} d'_{i,j} f'_{i,j} \quad (3.11)$$

s.a:

$$\left\{ \begin{array}{l} \sum_{(i,\ell) \in A'} f'_{i,\ell} - \sum_{(\ell,i) \in A'} f'_{\ell,i} = D(i) \quad \forall i \in U \quad (3.12) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{(\ell,n+1) \in A'} f'_{\ell,n+1} - \sum_{(n+1,\ell) \in A'} f'_{n+1,\ell} = K \quad (3.13) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{((n+1)',\ell) \in A'} f'_{(n+1)',\ell} - \sum_{(\ell,(n+1)') \in A'} f'_{\ell,(n+1)'} = K \quad (3.14) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{(I',\ell) \in A'} f'_{I',\ell} - \sum_{(\ell,I') \in A'} f'_{\ell,I'} = P - d^+(I) \quad (3.15) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{(\ell,I) \in A'} f'_{\ell,I} - \sum_{(I,\ell) \in A'} f'_{I,\ell} = P - d^-(I) \quad (3.16) \end{array} \right.$$

$$\left\{ \begin{array}{l} f'_{i,j} \geq 0 \text{ e inteiros} \quad \forall (i,j) \in A'. \quad (3.17) \end{array} \right.$$

Note-se que:

- $(n+1, (n+1)')$ é o único arco de A' com vértice

➤ inicial $n+1$. Logo, em (3.13), $\sum_{(n+1,\ell) \in A'} f'_{n+1,\ell} = f'_{n+1,(n+1)'} ;$

➤ final $(n+1)'$. Logo, em (3.14), $\sum_{(\ell,(n+1)') \in A'} f'_{\ell,(n+1)'} = f'_{n+1,(n+1)'} .$

- (I, I') é o único arco de A' com vértice

➤ inicial I . Logo, em (3.15), $\sum_{(\ell,I') \in A'} f'_{\ell,I'} = f'_{I,I'} ;$

➤ final I' . Logo, em (3.16), $\sum_{(I,\ell) \in A'} f'_{I,\ell} = f'_{I,I'} .$

Lema 3.1: Os problemas (B) e (B') são equivalentes.

Prova: Provando que:

- (a) qualquer solução admissível de (\mathbf{B}) pode ser transformada numa solução admissível de (\mathbf{B}') , com o mesmo valor;

e que:

- (b) qualquer solução admissível de (\mathbf{B}') pode ser transformada numa solução admissível de (\mathbf{B}) , com o mesmo valor;

fica demonstrado o lema.

- (a) Note-se que, qualquer solução admissível de (\mathbf{B}) verifica:

por (3.2') e (3.4''),

$$\sum_{(\ell, n+1) \in A} f_{\ell, n+1} = \sum_{(n+1, \ell) \in A} f_{n+1, \ell} = L_{n+1} + K, \quad \text{com } L_{n+1} \geq 0;$$

por (3.2') e (3.3'),

$$\sum_{(I, \ell) \in A} f_{I, \ell} + d^+(I) = \sum_{(\ell, I) \in A} f_{\ell, I} + d^-(I) = L_I + P, \quad \text{com } L_I \geq 0.$$

Seja $F = [f_{i,j}]$, $i, j \in V$ uma solução admissível de (\mathbf{B}) .

Uma solução admissível para (\mathbf{B}') , com mesmo valor da solução F , pode obter-se construindo $F' = [f'_{i,j}]$, $i, j \in V'$, da seguinte forma:

(i) $f'_{(n+1)', \ell} = f_{n+1, \ell}$, $\forall \ell \in V$

(ii) $f'_{n+1, (n+1)'} = L_{n+1}$

(iii) $f'_{I', \ell} = f_{I, \ell}$, $\forall \ell \in V$

(iv) $f'_{I, I'} = L_I$

(v) $f'_{i,j} = f_{i,j}$ para os restantes valores de $i, j \in V$: $(i, j) \in A'$.

- (b) Seja $F' = [f'_{i,j}]$, $i, j \in V'$ uma solução admissível de (\mathbf{B}') .

Construindo $F = [f_{i,j}]$, $i, j \in V$ da seguinte forma:

(i) $f_{n+1, \ell} = f'_{(n+1)', \ell}$, $\forall \ell \in V$

$$(ii) f_{1,\ell} = f'_{1',\ell}, \quad \forall \ell \in V$$

$$(iii) f_{i,j} = f'_{i',j} \text{ para os restantes valores de } i, j \in V: (i, j) \in A,$$

torna-se óbvio que F tem o mesmo valor de F' e que verifica as restrições (3.2') para todos os vértices $i \in U$, (3.3'), (3.4'') e (3.9').

Verificação das restrições (3.2') para $i = 1, n+1$:

- $i = n+1$

$$(3.13) \text{ e } (3.14) \Rightarrow$$

$$\begin{aligned} \Rightarrow \sum_{(\ell, n+1) \in A'} f'_{\ell, n+1} + \sum_{(\ell, (n+1)') \in A'} f'_{\ell, (n+1)'} &= \\ &= \sum_{(n+1, \ell) \in A'} f'_{n+1, \ell} + \sum_{((n+1)', \ell) \in A'} f'_{(n+1)', \ell} \Leftrightarrow \end{aligned}$$

$$\begin{aligned} \Leftrightarrow \sum_{(\ell, n+1) \in A'} f'_{\ell, n+1} + f'_{n+1, (n+1)'} &= \\ &= f'_{n+1, (n+1)'} + \sum_{((n+1)', \ell) \in A'} f'_{(n+1)', \ell} \Rightarrow \end{aligned}$$

$$\begin{aligned} \Rightarrow \sum_{(\ell, n+1) \in A} f_{\ell, n+1} &= \sum_{(n+1, \ell) \in A} f_{n+1, \ell} \\ \text{(i)} & \\ \text{(iii)} & \end{aligned}$$

\therefore como $D(n+1) = 0$, (3.2') é verificada para $i = n+1$.

- $i = 1$

$$(3.15) \text{ e } (3.16) \Rightarrow$$

$$\begin{aligned} \Rightarrow \sum_{(1', \ell) \in A'} f'_{1', \ell} + \sum_{(1, \ell) \in A'} f'_{1, \ell} + d^+(1) &= \\ &= \sum_{(\ell, 1') \in A'} f'_{\ell, 1'} + \sum_{(\ell, 1) \in A'} f'_{\ell, 1} + d^-(1) \Leftrightarrow \end{aligned}$$

$$\begin{aligned} \Leftrightarrow \sum_{(1', \ell) \in A'} f'_{1', \ell} + f'_{1, 1'} + d^+(1) &= \\ &= f'_{1, 1'} + \sum_{(\ell, 1) \in A'} f'_{\ell, 1} + d^-(1) \Rightarrow \end{aligned}$$

$$\begin{aligned} &\Rightarrow \sum_{(I,\ell) \in A} f_{I,\ell} + d^+(I) = \sum_{(\ell,I) \in A} f_{\ell,I} + d^-(I) \\ \text{(ii)} & \\ \text{(iii)} & \end{aligned}$$

\therefore (3.2') é verificada para $i = I$.

$\therefore F$ é uma solução admissível de (B) e tem o mesmo valor de F' . \square

O problema (B') pode, por sua vez, ser transformado num problema de transporte equivalente, considerando:

- como *conjunto de origens*: $O = \{I', (n+I)'\} \cup I$, sendo:

I' , cópia de I , uma origem de (B') , com oferta $a(I') = P - d^+(I)$, por (3.15).

$(n+I)'$, cópia de $n+I$, uma origem com oferta $a((n+I)') = K$, por (3.14).

$I = \{i \in U: D(i) > 0\}$ Por (3.12), estes vértices são origens de (B') , com ofertas $a(i) = D(i)$, e representam os vértices desequilibrados por terem grau interno superior ao grau externo, em G_R .

- como *conjunto de destinos*: $D = \{I, n+I\} \cup J$, onde:

I tem procura $b(I) = P - d^-(I)$. Pois, por (3.16), o vértice I é um destino.

$n+I$ tem procura $b(n+I) = K$. Por (3.13), $n+I$ representa um destino.

$J = \{j \in U: D(j) < 0\}$, com procuras $b(j) = -D(j)$, por (3.12).

- como *custos unitários de transporte* de cada origem para cada destino:

$$l_{i,j} = \begin{cases} scl(I) & \text{se } i = I' \wedge j = I \\ scl(n+I) & \text{se } i = (n+I)' \wedge j = n+I \\ spl(i,j) & \text{c.c.} \end{cases} \quad (3.18)$$

Nos lemas seguintes prova-se que os custos unitários de transporte de cada origem para cada destino, definidos por (3.18), são iguais aos comprimentos dos caminhos de menor custo dessa origem para esse destino em G' . De facto, neste caso a resolução de (B') equivale à resolução do problema de transporte definido por O, D e (3.18), o que é fácil de entender.

Lema 3.2: O comprimento do caminho de menor custo de I' para I , em G' , é igual ao comprimento do circuito de custo mínimo em vazio que contém I , em G .

Prova: Seja $SP(I', I)$ o caminho de custo mínimo de I' para I , em G' .

Sabendo que $(I', I) \notin A'$ então $SP(I', I)$ é formado por dois caminhos de menor custo. Seja $x \in V'$:

$$SP(I', I) = SP(I', x) \cup SP(x, I')$$

de comprimento: $spl(I', I) = spl(I', x) + spl(x, I)$.

Note-se que este comprimento coincide com o comprimento do caminho de menor custo em vazio pois, por hipótese, o custo de serviço associado a cada arco é sempre superior ao custo de percorrer esse mesmo arco em vazio.

Logo, em G , este custo é dado por $spl(I, x) + spl(x, I)$.

Por outro lado, em G , o circuito de custo mínimo que contém I e pelo menos mais um vértice do grafo, pode considerar-se como sendo formado por dois caminhos de custo mínimo em vazio.

Seja: $SC(I) = SP(I, y) \cup SP(y, I)$,

de custo: $scl(I) = spl(I, y) + spl(y, I)$.

Como é óbvio $spl(I, x) + spl(x, I) = spl(I, y) + spl(y, I)$,

caso contrário ou o circuito considerado em G não representaria o circuito de custo mínimo, ou o caminho considerado em G' não representaria o caminho de menor custo. \square

De forma análoga pode provar-se o lema referente aos custos associados aos arcos entre os vértices representativos da ETRS, que se enuncia de seguida.

Lema 3.3: O comprimento do caminho de menor custo de $(n+I)'$ para $n+I$, em G' , é igual ao comprimento do circuito de custo mínimo em vazio que contém $n+I$, em G . \square

Os custos unitários de transporte de cada origem para cada destino definidos por (3.18) são, como pretendido, iguais aos comprimentos dos caminhos mais curtos dessa origem para esse destino em G' .

Da formalização apresentada resultou então um subproblema (B') que pode ser resolvido pelo seguinte problema de transporte:

$$(PT1) \quad \text{Min } z_{PT1} = \sum_{j \in D} \sum_{i \in O} l_{i,j} t_{i,j}$$

s.a:

$$\left\{ \begin{array}{l} \sum_{j \in D} t_{i,j} = a(i) = \begin{cases} D(i) & \text{se } i \in U: D(i) > 0 \\ P - d^+(I) & \text{se } i = I' \\ K & \text{se } i = (n+1)' \end{cases} & i \in O \\ \sum_{i \in O} t_{i,j} = b(j) = \begin{cases} -D(j) & \text{se } j \in U: D(j) < 0 \\ P - d^-(I) & \text{se } j = I \\ K & \text{se } j = n+1 \end{cases} & j \in D \\ t_{i,j} \geq 0 & \forall i \in O, j \in D \end{array} \right.$$

As unidades a transportar da origem i para o destino j , $t_{i,j}$, representam no problema inicial PRRS:

- o número de vezes que é percorrido em vazio o circuito de custo mínimo que contém j , quando a origem e o destino são cópias do mesmo vértice inicial ($i = I' \wedge j = I$ ou $i = (n+1)' \wedge j = n+1$);
- ou o número de vezes que o caminho de menor custo em vazio da origem i para o destino j é percorrido, nos restantes casos,

como se ilustra de seguida.

Exemplo 3.3: Seja $P = 2$ e $W = 2$. Considere-se a seguinte representação do vértice $n+1 = 5$ e do grafo de procura, onde os números sobre os arcos indicam as respectivas procuras.

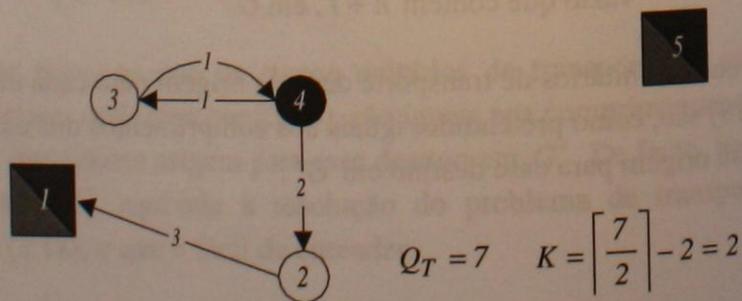


Figura 3.4: Grafo de procura e vértice $n+1$

O vértice: 1 é uma origem com oferta $P - d^+(1) = 2 - 0 = 2$

e um destino com procura $P - d^-(1) = 2 - 1 = 1$;

4 é um destino com procura $-D(4) = d^+(4) - d^-(4) = 1$;

$n+1 = 5$ é uma origem e um destino com oferta e procura $K = 2$.

Logo, a rede associada a **(PT1)** é:

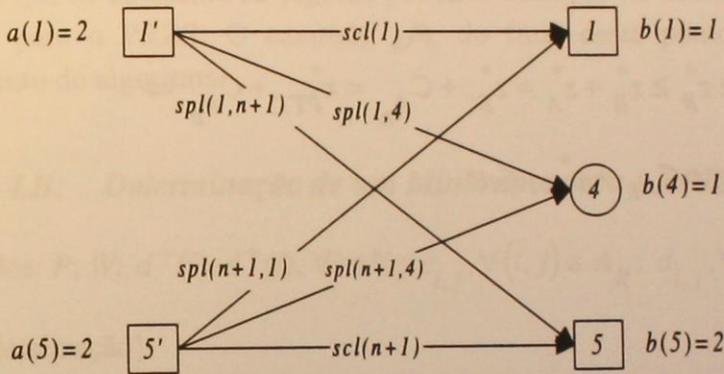


Figura 3.5: Rede associada a **(PT1)**

Seja:

- $t_{1',1} = 1$; $t_{1',4} = 1$ e $t_{5',5} = 2$, a solução ótima de **(PT1)**;
- $(1, 2, 1)$ o circuito de custo mínimo que contém 1 , em G ;
- $(5, 4, 5)$ o circuito de custo mínimo que contém 5 , em G ;
- $(1, 4)$ caminho de menor custo de 1 para 4 , em G ;

então, o multigrafo associado à solução anterior é o representado na figura seguinte

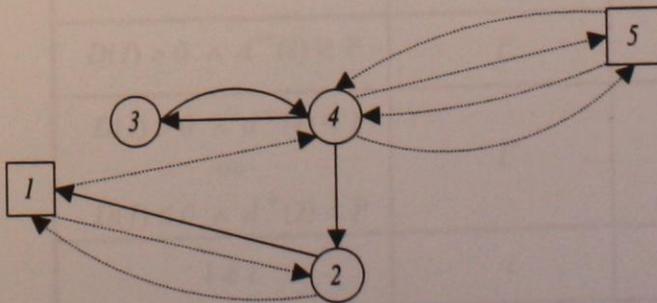


Figura 3.6: Multigrafo associado à solução de **(PT1)**

Esta solução nunca é pior que a solução que se obteria se fossem consideradas apenas as restrições de equilíbrio dos vértices. ◆

A construção dos subproblemas garante que a soma do custo de serviço com o valor óptimo do problema de transporte definido, representa um minorante válido para o valor óptimo de PRRS, como se estabelece na proposição seguinte.

Proposição 3.1: $\underline{z} = C_{AR} + z_{PTI}^*$ é um minorante para o valor óptimo do PRRS, ou seja:

$$\underline{z} \leq z^*.$$

Prova: $z^* \geq z_R^* \geq z_B^* + z_A^* = z_{B'}^* + C_{AR} = z_{PTI}^* + C_{AR} \Rightarrow$
 $\Rightarrow \underline{z} = C_{AR} + z_{PTI}^* \leq z^*.$ □

3.3. Algoritmo para a Determinação de Minorantes

A determinação de minorantes para o PRRS pode então basear-se na relaxação apresentada. Como se provou ser possível determinar minorantes para o PRRS através da resolução do problema de transporte e do cálculo do custo de serviço, o algoritmo que se apresenta de seguida pode ser utilizado na determinação de minorantes para o PRRS. O exemplo **A**, do final deste ponto, ilustra o funcionamento do algoritmo.

Algoritmo LB: Determinação de um Minorante para o PRRS

0. Dados: P ; W ; $d^-(i), d^+(i), \forall i \in V$; $c_{i,j}, \forall (i,j) \in A_R$; $d_{i,j}, \forall (i,j) \in A$;
1. {Inicialização}
 - 1.1 Calcular: Q_T ; K ; $D(i), \forall i \in V$; C_{A_R} ;
 - 1.2 Determinar o caminho de menor custo em vazio entre qualquer par de vértices;
 - 1.3 Determinar o circuito de custo mínimo em vazio que contém l ;
 - 1.4 Determinar o circuito de custo mínimo em vazio que contém $n+l$.
2. {Construção da rede para o problema de transporte}
 - 2.1 Definir as origens e respectivas ofertas por:

Se	Origem ($i \in O$)	Oferta: $a(i)$
	$(n+l)'$	K
$D(l) > 0 \wedge d^-(l) \geq P$	l'	$D(l)$
$D(l) > 0 \wedge d^-(l) < P$ ou $D(l) \leq 0 \wedge d^+(l) < P$	l'	$P - d^+(l)$
$i \in l$	i	$D(i)$

Tabela 3.2: Origens de (PT1)

2.2 Definir os destinos e respectivas procuras por:

Se	Destino ($j \in D$)	Procura: $b(j)$
	$n+1$	K
$D(l) < 0 \wedge d^+(l) \geq P$	l	$-D(l)$
$D(l) < 0 \wedge d^+(l) < P$ ou $D(l) \geq 0 \wedge d^-(l) < P$	l	$P - d^-(l)$
$j \in J$	j	$-D(j)$

Tabela 3.3: Destinos de (PT1)

2.3 Definir os custos unitários de transporte de cada origem ($i \in O$) para cada destino ($j \in D$) de acordo com (3.18).

3. Determinar a solução óptima do problema de transporte definido em 2. e o respectivo valor óptimo, z_{PT1}^* .

4. Calcular o valor do minorante, $\underline{z} = C_{AR} + z_{PT1}^*$.

Fim.

No exemplo que se apresenta de seguida, ilustra-se o funcionamento deste algoritmo.

Exemplo A: Suponha-se um problema com as seguintes características:

- $n+1 = 11$ número de vértices;
- $P = 2$; número de veículos;
- $W = 4$; capacidade de cada veículo;
- $q_{i,j} = 1, \forall (i,j) \in A_R$; procuras nos arcos.

Considere-se a seguinte representação do grafo de procura conjuntamente com o vértice $n+1$:

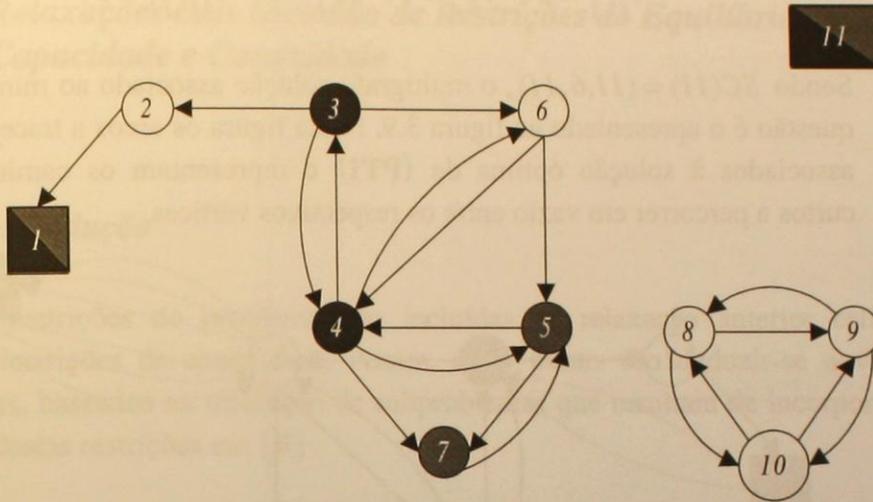


Figura 3.7: Grafo de procura e $n+1=11$

$$\text{Logo, } \left\lfloor \frac{Q_T}{W} \right\rfloor = \left\lfloor \frac{19}{4} \right\rfloor = 5 \Rightarrow K = 5 - 2 = 3.$$

A rede associada a (PT1) é ⁽¹⁾:

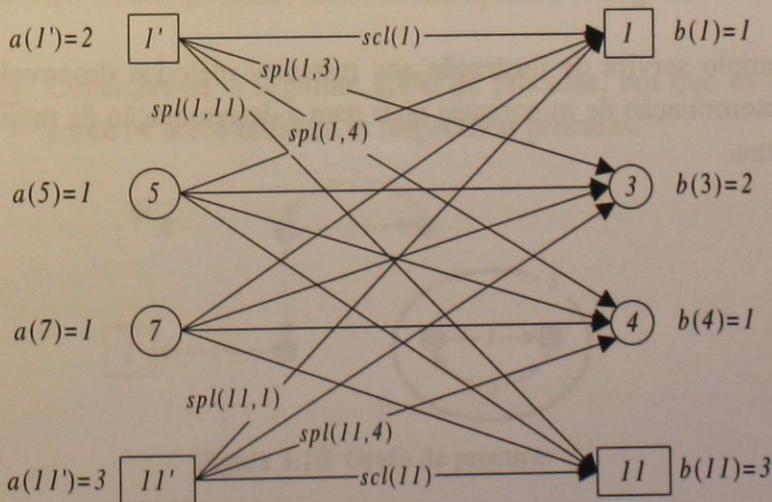


Figura 3.8: Rede associada a (PT1)

⁽¹⁾ Incluem-se apenas alguns dos custos para não sobrecarregar a representação. Os custos associados a cada arco são sempre iguais aos comprimentos dos caminhos de custo mínimo das origens para os destinos.

Considere-se que a solução óptima de (PT1) é:

$$t_{1',3} = 2; \quad t_{5,4} = 1; \quad t_{7,1} = 1; \quad t_{11',11} = 3.$$

Sendo $SC(11) = (11,6,11)$, o multigrafo solução associado ao minorante em questão é o apresentado na figura 3.9. Nesta figura os arcos a tracejado estão associados à solução óptima de (PT1) e representam os caminhos mais curtos a percorrer em vazio entre os respectivos vértices.

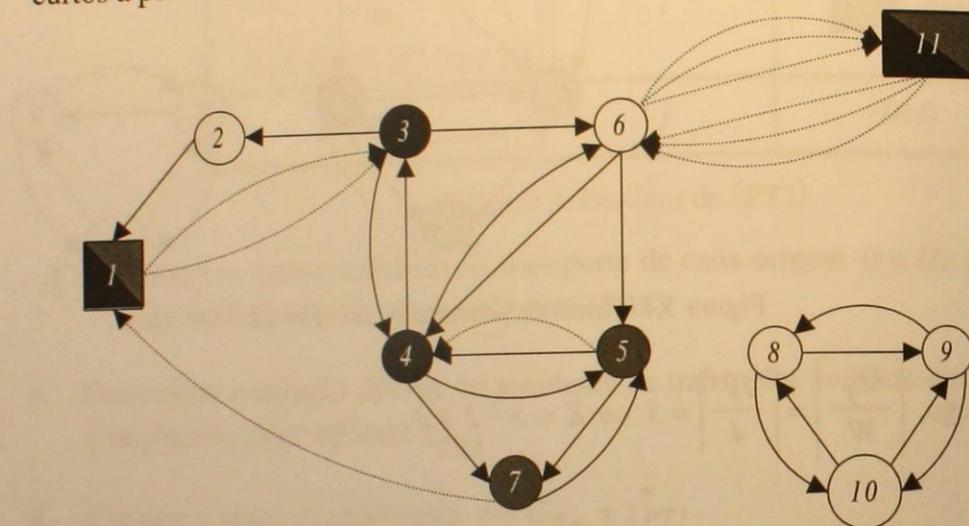


Figura 3.9: Multigrafo associado à solução de LB

Este exemplo servirá de ilustração aos restantes métodos desenvolvidos, quer para a determinação de minorantes quer para a determinação de majorantes para o problema.

3.4. Relaxações com Inclusão de Restrições de Equilíbrio, Capacidade e Conexidade

3.4.1. Introdução

Entre as restrições do problema não incluídas na relaxação anterior estão algumas restrições de conexidade. Assim, neste ponto vão deduzir-se novos minorantes, baseados na resolução de subproblemas que resultam de incorporar algumas destas restrições em **(B)**.

Se o grafo de procuras contiver diferentes componentes conexas equilibradas (ou seja, em que todos os vértices tenham grau interno igual ao grau externo) pode ser definido um novo subproblema cuja solução inclua a ligação de cada componente conexa equilibrada ou a $n+1$ e/ou a restantes vértices do grafo. Para tal, considera-se em **(B)** uma restrição associada a cada uma dessas componentes. Com estas restrições tenta garantir-se que o número total de arcos que devem incidir em vértices da componente é pelo menos suficiente para efectuar o serviço na componente, como se exemplifica de seguida.

Exemplo 3.4: Considere-se o seguinte grafo de procuras, em que os números sobre os arcos indicam as respectivas procuras.

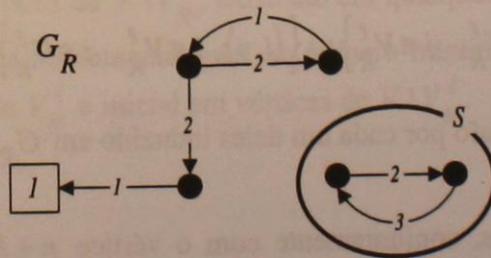


Figura 3.10: Grafo de procuras

Suponha-se que a capacidade dos veículos disponíveis é $W = 4$.

Em qualquer solução admissível do problema terão que ser percorridos, em vazio, dois arcos com extremo final em vértices de S e dois arcos com extremo inicial em vértices de S , uma vez que a procura total nesta componente é 5, superior à capacidade dos veículos. Assim, pretendem incluir-se novas restrições na relaxação que representem a possibilidade de efectuar o serviço na componente S . ♦

No problema inicial, definido em (2.2)–(2.15), as restrições de conexidade (2.11), foram escritas em função da existência de arcos de procura ou de cópias de arcos de procura nula em cada possível subgrafo. Podiam, porém, ter sido escritas tendo em conta a procura total em cada subgrafo. Neste caso, e considerando a agregação das restrições para o conjunto das subrotas de cada rota, viria:

$$\sum_{i \in S} \sum_{j \in S} q_{i,j} x_{i,j}^p \leq W \sum_{i \in S} \sum_{j \in S} (x_{i,j}^p + y_{i,j}^p) \quad \forall S \subseteq V \setminus \{n+1\}; p = 1, \dots, P. \quad (2.11')$$

Somando em p e fazendo, como anteriormente, $f_{i,j} = \sum_{p=1}^P y_{i,j}^p$, resulta:

$$Q_S \leq W \sum_{i \in S} \sum_{j \in S} \left(\sum_{p=1}^P x_{i,j}^p + f_{i,j} \right) \quad \forall S \subseteq V \setminus \{n+1\}, \quad (3.19)$$

onde $Q_S = \sum_{i \in S} \sum_{j \in S} q_{i,j} x_{i,j}^p$ representa a quantidade total de resíduos existentes no subgrafo induzido pelo conjunto de vértices S .

Considere-se que existem nc subconjuntos de vértices $V_R^\ell \subseteq V_R$, tais que:

H1) $\forall i \in V_R^\ell: d^-(i) = d^+(i)$, ou seja, todos os vértices de V_R^ℓ são equilibrados;

H2) $\left[\left\{ (i,j): i \in V_R^\ell, j \notin V_R^\ell \right\} \cup \left\{ (i,j): i \notin V_R^\ell, j \in V_R^\ell \right\} \right] \cap A_R = \emptyset$;

H3) e que o subgrafo por cada um deles induzido em G_R é conexo.

O subgrafo de procuras, conjuntamente com o vértice $n+1$, pode então ser escrito da seguinte forma:

$$\bigcup_{\ell=1}^{nc} G_R^\ell \cup \bar{G}_R,$$

onde: $G_R^\ell = (V_R^\ell, A_R^\ell)$ representam as diferentes componentes conexas equilibradas de G_R ;

$$\bar{G}_R = (\bar{V}, \bar{A}_R) \quad \text{com} \quad \bar{V} = V \setminus \bigcup_{\ell=1}^{nc} V_R^\ell \quad (1)$$

e $\bar{A}_R = A_R \setminus \bigcup_{\ell=1}^{nc} A_R^\ell$, representando os arcos de procura entre vértices de \bar{V} .

Por **H2)**,
$$\sum_{i \in V_R^\ell} \sum_{j \in V_R^\ell} \sum_{p=1}^P x_{i,j}^p = 0, \quad \forall \ell = 1, \dots, nc.$$

Se $Q_\ell = \sum_{a \in A_R^\ell} q_a$ for a quantidade total de resíduos na ℓ -ésima componente

conexa, (3.19) vem:

$$Q_\ell \leq W \sum_{i \in V_R^\ell} \sum_{j \in V_R^\ell} f_{i,j} \Leftrightarrow \sum_{i \in V_R^\ell} \sum_{j \in V_R^\ell} f_{i,j} \geq \left\lceil \frac{Q_\ell}{W} \right\rceil \quad \forall \ell = 1, \dots, nc. \quad (3.20)$$

Defina-se para cada componente conexa equilibrada, ou seja, $\forall \ell = 1, \dots, nc$,

$$K_\ell \equiv \left\lceil \frac{Q_\ell}{W} \right\rceil. \quad (3.21)$$

K_ℓ representa o número mínimo de arcos com extremo inicial em vértices de V_R^ℓ e final em vértices de $V \setminus V_R^\ell$, existente em qualquer solução admissível do PRRS. Este valor é obviamente igual ao número mínimo de arcos com vértice final em vértices de V_R^ℓ e inicial em vértices de $V \setminus V_R^\ell$.

(1) Note-se que $l \in \bar{V}$, pois está-se a supor que $d^+(l) < d^-(l) < P$.

Em vez do subproblema (B) considere-se então:

$$(B1) \text{ Min } z_{B1} = \sum_{(i,j) \in A} d_{i,j} f_{i,j} \quad (3.1.B1)$$

s.a :

$$\left\{ \begin{array}{l} \sum_{(i,\ell) \in A} f_{i,\ell} - \sum_{(\ell,i) \in A} f_{\ell,i} = D(i) \quad \forall i \in V \quad (3.2') \\ \sum_{(l,\ell) \in A} f_{l,\ell} \geq P - d^+(l) \quad (3.3') \\ \sum_{(n+1,\ell) \in A} f_{n+1,\ell} \geq K \quad (3.4'') \\ \sum_{i \in V_R^\ell} \sum_{\substack{(i,j) \in A \\ j \in V_R^\ell}} f_{i,j} \geq K_\ell \quad \ell = 1, \dots, nc \quad (3.20) \\ f_{i,j} \geq 0 \text{ e inteiros} \quad \forall (i,j) \in A. \quad (3.9') \end{array} \right.$$

As restrições (3.20), aparecem pela primeira vez, e impõem que o número mínimo de ligações com cada uma destas componentes conexas equilibradas seja suficiente para que o serviço possa ser realizado, tendo em conta a capacidade dos veículos.

Como é óbvio, a soma dos valores óptimos dos subproblemas (B1) e (A) proporcionam ainda um minorante válido para o PRRS, sendo: $z_{B1}^* \geq z_B^*$.

Porém, não se consegue transformar (B1) num problema de simples resolução sem recorrer a relaxações. Vão então, de seguida, construir-se duas relaxações deste problema, (B2) e (B3), resolúveis em tempo polinomial. Tratando-se de relaxações de (B1) conduzem, obviamente, conjuntamente com (A), a minorantes válidos para o problema inicial, PRRS.

A relaxação (B2) designa-se por relaxação com inclusão de super-vértices, pois cada um dos nc subconjuntos de vértices V_R^ℓ é representado por um super-vértice. À semelhança do processo utilizado na relaxação anterior, consegue assim obter-se um problema resolúvel por um problema de transporte, de valor óptimo z_{PT2}^* . Neste problema de transporte, os super-vértices definidos representam novas origens e novos destinos. Porém, desta forma perde-se a informação associada a cada um dos vértices em cada uma das componentes conexas equilibradas, podendo obter-se soluções em que alguns

dos vértices destas componentes resultem não equilibrados. No entanto, passa a ser possível ligar os vértices em cada um dos conjuntos V_R^ℓ com os restantes vértices do grafo, com pelo menos K_ℓ arcos.

A relaxação (B3) é designada por relaxação com duplicação de destinos. Nesta é mantida a estrutura inicial dos vértices dos conjuntos V_R^ℓ ($\ell = 1, \dots, nc$), não desequilibrando vértices inicialmente equilibrados. Para tal, e tendo em conta que (B1) representa um problema de fluxos com três conjuntos de restrições adicionais, duplicam-se todos os destinos e considera-se que o fluxo de uma origem para um destino pode passar directamente dessa origem para esse destino, ou via as componentes conexas equilibradas, considerando todas as hipóteses possíveis. Obtém-se, neste caso, um problema de fluxo de custo mínimo, com valor óptimo z_{B3}^* .

3.4.2. Super-Vértices

Na tentativa de exigir que no multigrafo associado à solução do método para determinar minorantes exista um certo número mínimo de arcos de procura nula incidentes em cada componente conexa equilibrada vai representar-se cada um dos nc subconjuntos de vértices V_R^ℓ por um super-vértice.

Defina-se então o grafo $\hat{G} = (\hat{V}, \hat{A})$, onde:

- o conjunto de vértices é $\hat{V} = \bar{V} \cup S$, com

$$\bar{V} = V \setminus \bigcup_{\ell=1}^{nc} V_R^\ell \quad \text{e} \quad S = \{s_1, s_2, \dots, s_{nc}\}.$$

Cada nodo s_ℓ representa uma das nc componentes conexas equilibradas.

- o conjunto de arcos, \hat{A} , representa as ligações iniciais entre os vértices de \bar{V} e entre estes e vértices de cada uma das componentes conexas equilibradas, agora representadas por super-vértices.

Assim, define-se \hat{A} por ⁽¹⁾:

$$\begin{cases} (i, j) \in \hat{A} & \text{se } (i, j) \in A \wedge i, j \in \bar{V}; \\ (i, s_j) \in \hat{A} & \text{se } i \in \bar{V} \wedge \exists v \in V_R^j: (i, v) \in A; \\ (s_i, j) \in \hat{A} & \text{se } \exists u \in V_R^i: (u, j) \in A \wedge j \in \bar{V}; \\ (s_i, s_j) \in \hat{A} & \text{se } \exists u \in V_R^i; v \in V_R^j: (u, v) \in A. \end{cases}$$

Os custos associados a cada arco $(i, j) \in \hat{A}$, com $i \neq j$, são dados por:

$$\hat{d}_{i,j} = \begin{cases} d_{i,j} & \text{se } i, j \in \bar{V}; \\ \text{Min}_{v \in V_R^j} \{spl(i, v)\} & \text{se } i \in \bar{V} \wedge j = s_j \in S; \\ \text{Min}_{u \in V_R^i} \{spl(u, j)\} & \text{se } i = s_i \in S \wedge j \in \bar{V}; \\ \text{Min}_{\substack{u \in V_R^i \\ v \in V_R^j}} \{spl(u, v)\} & \text{se } i = s_i, j = s_j \in S. \end{cases}$$

⁽¹⁾ Os arcos repetidos nos dois conjuntos, A e \hat{A} , têm as mesmas características. Ou seja, têm os mesmos custos (de serviço e de atravessar em vazio) e procuras associadas.

Exemplo A: Considere-se de novo o exemplo **A**, em que o grafo original está representado na figura seguinte:

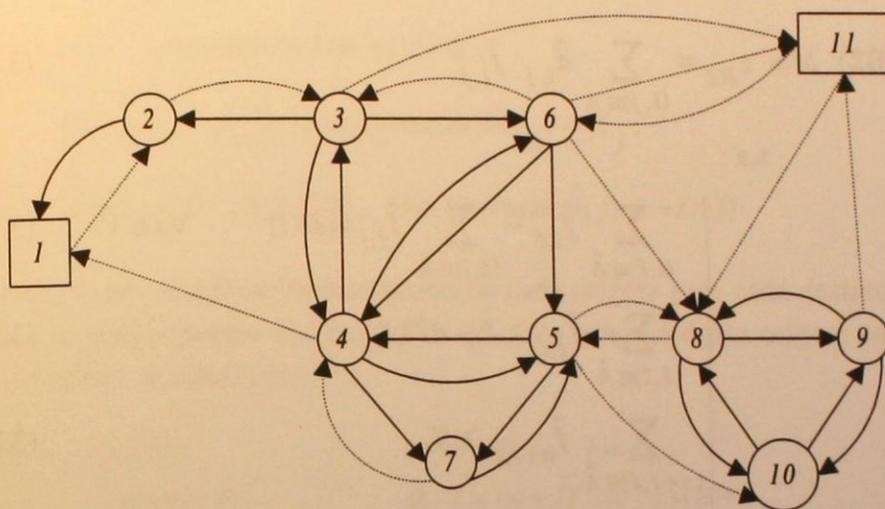


Figura 3.11: Grafo G

Neste caso, \hat{G} representa-se por:

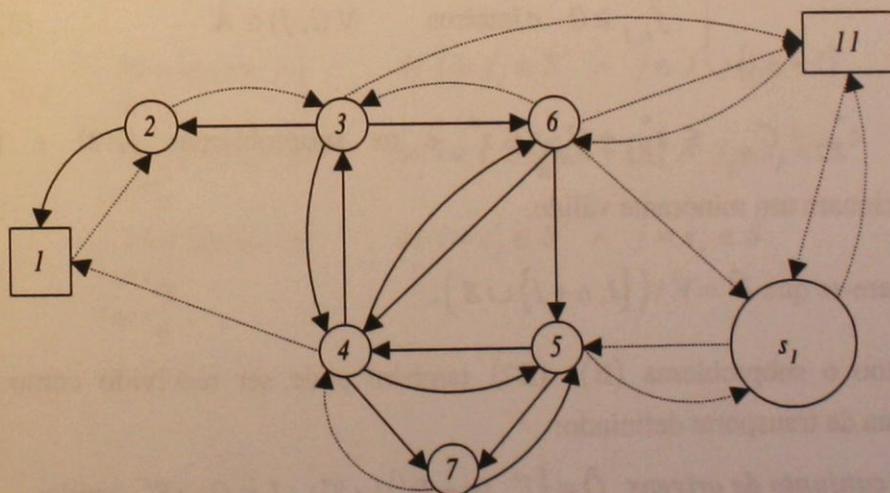


Figura 3.12: Grafo \hat{G}

Sendo $S = \{s_1\}$; $V_R^I = \{8, 9, 10\}$; $K_{s_1} = \left\lceil \frac{6}{4} \right\rceil = 2$;

$$\hat{d}_{s_1, 11} = d_{9, 11}; \quad \hat{d}_{s_1, 5} = d_{8, 5}; \quad \hat{d}_{6, s_1} = d_{6, 8}$$

$$\hat{d}_{5, s_1} = \text{Min}\{d_{5, 8}; d_{5, 10}\} = d_{5, 8} \quad \text{e} \quad \hat{d}_{11, s_1} = d_{11, 8} \quad \blacklozenge$$

Representando por $\hat{f}_{i,j}$ o número de vezes que o arco $(i,j) \in \hat{A}$ é percorrido em vazio, pelos P veículos, obtém-se:

$$(B2) \text{ Min } z_{B2} = \sum_{(i,j) \in \hat{A}} \hat{d}_{i,j} \hat{f}_{i,j} \quad (3.22)$$

s.a :

$$\left\{ \begin{array}{l} \sum_{(i,\ell) \in \hat{A}} \hat{f}_{i,\ell} - \sum_{(\ell,i) \in \hat{A}} \hat{f}_{\ell,i} = D(i) \quad \forall i \in \hat{V} \end{array} \right. \quad (3.23)$$

$$\left\{ \begin{array}{l} \sum_{(I,\ell) \in \hat{A}} \hat{f}_{I,\ell} \geq P - d^+(I) \end{array} \right. \quad (3.24)$$

$$\left\{ \begin{array}{l} \sum_{(n+1,\ell) \in \hat{A}} \hat{f}_{n+1,\ell} \geq K \end{array} \right. \quad (3.25)$$

$$\left\{ \begin{array}{l} \sum_{(\ell,j) \in \hat{A}} \hat{f}_{\ell,j} \geq K_\ell \quad \forall s_\ell \in S \end{array} \right. \quad (3.26)$$

$$\left\{ \begin{array}{l} \hat{f}_{i,j} \geq 0 \text{ e inteiros} \quad \forall (i,j) \in \hat{A}. \end{array} \right. \quad (3.27)$$

Assim, $z_{B2}^* + C_{A_R} \leq z_{BI}^* + C_{A_R} \leq z^*$ e os subproblemas (B2) e (A) proporcionam um minorante válido.

Considere-se que $\hat{U} = \hat{V} \setminus (\{I, n+1\} \cup S)$.

Tal como o subproblema (B), (B2) também pode ser resolvido como um problema de transporte definindo:

- o conjunto de origens $\hat{O} = \{I', (n+1)'\} \cup S' \cup I = O \cup S'$, sendo:

I' , uma origem com oferta $\hat{a}(I') = P - d^+(I)$;

$(n+1)'$, uma origem com oferta $\hat{a}((n+1)') = K$;

$S' = \{s'_1, \dots, s'_{nc}\}$ origens que representam cópias dos super-vértices associados às componentes conexas equilibradas. A oferta de cada s'_i é $\hat{a}(s'_i) = K_i$;

$I = \{i \in \hat{U} : D(i) > 0\}$ origens, com ofertas $\hat{a}(i) = D(i)$, que representam os vértices desequilibrados por terem grau interno superior ao grau externo, em G_R .

- o conjunto de destinos $\hat{D} = \{1, n+1\} \cup S \cup J = D \cup S$, onde:

1 tem procura $\hat{b}(1) = P - d^-(1)$;

$n+1$ tem procura $\hat{b}(n+1) = K$;

$S = \{s_1, \dots, s_{nc}\}$, com procuras $\hat{b}(s_j) = K_j$;

$J = \{j \in \hat{U}: D(j) < 0\}$, com procuras $\hat{b}(j) = -D(j)$.

- os custos unitários de transporte de cada origem para cada destino iguais aos comprimentos dos caminhos de custo mínimo dessa origem para esse destino, são dados por ⁽¹⁾:

$$\hat{l}_{i,j} = \begin{cases} scl(1) & \text{se } i=1' \wedge j=1 \\ scl(n+1) & \text{se } i=(n+1)' \wedge j=n+1 \\ scl(s_\ell) & \text{se } i=s'_\ell \in S' \wedge j=s_\ell \in S \\ spl(i,j) & \text{se } i \in I \wedge j \in J \\ \text{Min}_{u \in V_R^\ell} \{spl(u,j)\} & \text{se } i=s'_\ell \in S' \wedge j \in J \cup \{1, n+1\} \\ \text{Min}_{v \in V_R^\ell} \{spl(i,v)\} & \text{se } i \in I \cup \{1', (n+1)'\} \wedge j=s_\ell \in S \\ \text{Min}_{\substack{u \in V_R^\ell \\ v \in V_R^k}} \{spl(u,v)\} & \text{se } i=s'_\ell \in S' \wedge j=s_k \in S \end{cases} \quad (3.28)$$

⁽¹⁾ De forma semelhante ao que foi feito na relaxação anterior pode provar-se que o custo dos caminhos de menor custo de s'_ℓ para s_ℓ , de $1'$ para 1 e de $(n+1)'$ para $n+1$ são iguais aos custos dos circuitos de custo mínimo em vazio que contêm s_ℓ , 1 e $n+1$ em \hat{G} , respectivamente.

Lema 3.4: (B2) é equivalente ao seguinte problema de transporte:

$$(PT2) \quad \text{Min } z_{PT2} = \sum_{j \in \hat{D}} \sum_{i \in \hat{O}} \hat{t}_{i,j} \hat{t}_{i,j}$$

s.a :

$$\begin{cases} \sum_{j \in \hat{D}} \hat{t}_{i,j} = \hat{a}(i) & \forall i \in \hat{O} \\ \sum_{i \in \hat{O}} \hat{t}_{i,j} = \hat{b}(j) & \forall j \in \hat{D} \\ \hat{t}_{i,j} \geq 0 & \forall i \in \hat{O}, j \in \hat{D}. \end{cases}$$

□

As unidades a transportar da origem i para o destino j , $\hat{t}_{i,j}$, representam no problema inicial, PRRS:

- se $i = l' \wedge j = l$ ou $i = (n+1)' \wedge j = n+1$
o número de vezes que é percorrido em vazio o circuito de custo mínimo que contém j ;
- se $i = s'_\ell \in S' \wedge j = s_\ell \in S$
o número de vezes que são percorridos em vazio caminhos de custo mínimo entre pares de vértices de V_R^ℓ contendo pelo menos um vértice de $V \setminus V_R^\ell$;
- se $i \in I \wedge j \in J$
o número de vezes que o caminho de menor custo em vazio da origem i para o destino j é percorrido;
- se $i = s'_\ell \in S' \wedge j \in J \cup \{l, n+1\}$
o número de vezes que o caminho de custo mínimo em vazio do vértice de V_R^ℓ mais próximo de j para o destino j é percorrido;
- se $i \in I \cup \{l', (n+1)'\} \wedge j = s_\ell \in S$
o número de vezes que o caminho de menor custo em vazio da origem i para o vértice de V_R^ℓ mais próximo de i é percorrido;

- se $i = s'_\ell \in S \wedge j = s_k \in S$

o número de vezes que é percorrido em vazio o caminho de custo mínimo do vértice de V_R^ℓ mais próximo de V_R^k para V_R^k .

A prova do lema 3.4 passa pela transformação de **(B2)** num problema de fluxo de custo mínimo equivalente, **(B2')**, tal como foi feito anteriormente ao se transformar **(B)** em **(B')**.

A construção de **(B2')** baseia-se também na definição de um novo grafo $\hat{G}' = (\hat{V}', \hat{A}')$, onde:

- $\hat{V}' = \hat{V} \cup \{I', (n+1)', s'_1, s'_2, \dots, s'_{nc}\}$;
- $\hat{A}' = \left[\hat{A} \setminus \left(\hat{A}_{n+1}^+ \cup \hat{A}_{I'}^+ \cup \bigcup_{\ell=1}^{nc} \hat{A}_{s'_\ell}^+ \right) \right] \cup \{(n+1, (n+1)'), (I, I')\} \cup \bigcup_{\ell=1}^{nc} \{(s_\ell, s'_\ell)\} \cup \hat{A}_{(n+1)'}^+ \cup \hat{A}_{I'}^+ \cup \bigcup_{\ell=1}^{nc} \hat{A}_{s'_\ell}^+$,

com $\hat{A}_k^+ = \{(k, j) \in \hat{A} : j \in \hat{V}\}$ e $\hat{A}_{k'}^+ = \{(k', j) : (k, j) \in \hat{A}, j \in \hat{V}\}$.

Analogamente, os custos unitários associados aos arcos de \hat{A}' devem ser definidos por:

$$\hat{d}'_{i,j} = \begin{cases} \hat{d}_{u,v} & \text{se } (i, j) \in \hat{A}' \text{ resulta de } (u, v) \in \hat{A} \\ 0 & \text{c.c.} \end{cases}$$

É então possível provar que o valor óptimo de **(PT2)** somado com o custo de serviço fornece um minorante válido para o valor óptimo do PRRS, como se estabelece de seguida.

Proposição 3.2: $z_2 = C_{A_R} + z_{PT2}^*$ é um minorante para o valor óptimo do PRRS, ou seja:

$$z_2 \leq z^*.$$

Prova: $z^* \geq z_{B1}^* + C_{A_R} \geq z_{B2}^* + C_{A_R} = z_{PT2}^* + C_{A_R} \Rightarrow$
 $\Rightarrow z_2 = C_{A_R} + z_{PT2}^* \leq z^*.$ □

Exemplo A: Para este exemplo a rede associada a (PT2) é:

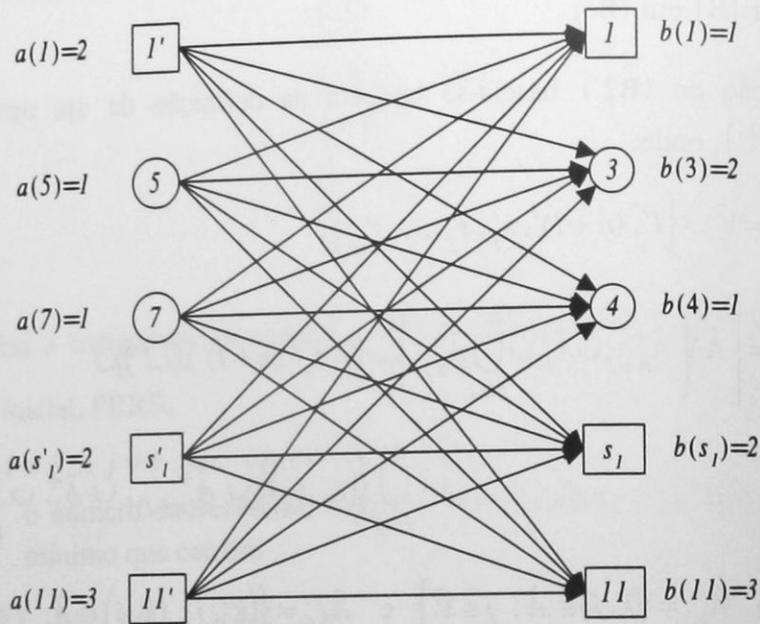


Figura 3.13: Rede associada a (PT2)

Onde, por exemplo,

$$\hat{l}_{s'_1,4} = \text{Min}_{i \in V_R^1} \{spl(i,4)\} = spl(8,4);$$

$$\hat{l}_{11',s_1} = \text{Min}_{j \in V_R^1} \{spl(11,j)\} = spl(11,8);$$

$$\hat{l}_{s'_1,s_1} = scl(s_1) = spl(8,5) + spl(5,8).$$

Considere-se que a solução óptima de (PT2) é:

$$t_{1',3} = 2; t_{7,1} = 1; t_{5,s_1} = 1; t_{s'_1,4} = 1; t_{s'_1,11} = 1; t_{11',s_1} = 1; t_{11',11} = 2.$$

O multigrafo solução associado ao minorante em questão é o da figura 3.14. Nesta figura os arcos a tracejado estão associados à solução óptima de (PT2) e representam os caminhos de custo mínimo a percorrer em vazio entre os respectivos vértices.

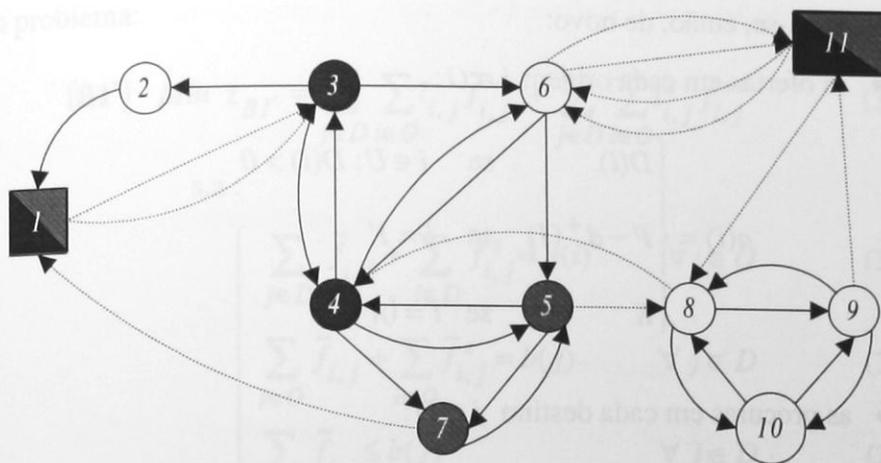


Figura 3.14: Multigrafo associado à solução de (PT2)

Neste multigrafo, embora existam arcos de procura nula a incidir em vértices de V_R^I , os vértices 8 e 9 estão desequilibrados, tendo em conta todos os arcos representados. ◆

Assim, o multigrafo solução resultante deste minorante pode, como se verifica neste exemplo, resultar não Euleriano.

3.4.3. Duplicação de Destinos

Considere-se, antes de mais, que existe apenas uma componente conexa equilibrada, $nc = 1$, e represente-se por s . Seja V_R^S o conjunto de vértices na componente conexa.

Neste ponto constrói-se uma nova relaxação na tentativa de manter o equilíbrio dos vértices da componente conexa equilibrada ao exigir a existência de arcos incidentes em vértices de V_R^S . Assim vão duplicar-se todos os destinos de (B1)⁽¹⁾ e considerar que os arcos que ligam as origens aos destinos duplicados representam um caminho que passa por vértices de V_R^S .

Vão ainda ser introduzidos dois conjuntos de restrições que embora inicialmente redundantes, o deixam de ser no problema final que se obterá, fortalecendo portanto o valor do minorante.

(1) Problema definido na página 58.

Considerem-se, então, de novo:

- as ofertas em cada origem $i \in O$:

$$a(i) = \begin{cases} D(i) & \text{se } i \in U: D(i) > 0 \\ P - d^+(I) & \text{se } i = I' \\ K & \text{se } i = (n+1)' \end{cases}$$

- as procuras em cada destino $j \in D$:

$$b(j) = \begin{cases} -D(j) & \text{se } j \in U: D(j) < 0 \\ P - d^-(I) & \text{se } j = I \\ K & \text{se } j = n+1 \end{cases}$$

Definam-se as variáveis:

- $\tilde{f}_{i,j}$ representando o número de vezes que é percorrido em vazio o caminho de custo mínimo de i para j . Como anteriormente, este caminho tem custo $l_{i,j}$ definido, por (3.18);

- $\tilde{f}_{i,j}^s$ representando o número de vezes que é percorrido em vazio o caminho de custo mínimo de i para j , que intersecta V_R^s . O custo é dado por $l_{i,j}^s = \text{Min}_{v \in V_R^s} \{ \text{spl}(i,v) + \text{spl}(v,j) \}$.

Seja o problema:

$$(B1') \quad \text{Min } z_{B1'} = \sum_{j \in D} \sum_{i \in O} l_{i,j} \tilde{f}_{i,j} + \sum_{j \in D} \sum_{i \in O} l_{i,j}^s \tilde{f}_{i,j}^s \quad (3.29)$$

s.a :

$$\left\{ \begin{array}{l} \sum_{j \in D} \tilde{f}_{i,j} + \sum_{j \in D} \tilde{f}_{i,j}^s = a(i) \quad \forall i \in O \quad (3.30) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i \in O} \tilde{f}_{i,j} + \sum_{i \in O} \tilde{f}_{i,j}^s = b(j) \quad \forall j \in D \quad (3.31) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i \in O} \tilde{f}_{i,j} \leq b(j) \quad \forall j \in D \quad (3.32) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i \in O} \tilde{f}_{i,j}^s \leq b(j) \quad \forall j \in D \quad (3.33) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{j \in D} \sum_{i \in O} \tilde{f}_{i,j}^s \geq K_s \quad (3.34) \end{array} \right.$$

$$\left\{ \begin{array}{l} \tilde{f}_{i,j} \geq 0 \text{ e inteiros} \quad \forall i \in O; \forall j \in D \quad (3.35) \end{array} \right.$$

$$\left\{ \begin{array}{l} \tilde{f}_{i,j}^s \geq 0 \text{ e inteiros} \quad \forall i \in O; \forall j \in D. \quad (3.36) \end{array} \right.$$

A componente conexa equilibrada, s , pode ser encarada como ponto obrigatório de passagem para K_s unidades do fluxo a enviar na rede das origens de O para os destinos de D .

As restrições redundantes, introduzidas de novo em (3.32) e (3.33), deixarão de o ser quando forem relaxadas as condições (3.31).

Lema 3.5: Os problemas (B1) e (B1') são equivalentes.

Prova: Seja: f^* uma solução óptima de (B1), de valor óptimo z_{B1}^* .

\tilde{f}^* uma solução óptima de (B1'), de valor óptimo $z_{B1'}^*$.

Provando ser possível construir:

a) uma solução admissível para (B1'), a partir de f^* , de valor igual

a z_{B1}^* ,

b) e uma solução admissível para (B1), a partir de \tilde{f}^* , de valor

igual a $z_{B1'}^*$,

fica demonstrado o lema.

a) Algoritmo: Construção de uma solução admissível para (B1') a partir de f^*

1. $\forall i \in O, j \in D: \tilde{f}_{i,j} = 0; \tilde{f}_{i,j}^s = 0;$

2. Se existirem vértices de V_R^s por analisar:
considerar um qualquer, seja $v \in V_R^s;$

Caso contrário, ir para 3.;

2.1. Considerar uma variável não nula $f_{u,v}^* > 0;$

2.2. Encontrar, na solução ótima de (B1) um caminho maximal em vazio que contenha (u,v) . Seja:
 $P(i,j) = (i, \dots, u, v, \dots, j);$

Então, $i \in O$ e $j \in D;$

2.3. Retirar o caminho encontrado da solução ótima, fazendo:

$$f_{\alpha,\beta}^* \leftarrow f_{\alpha,\beta}^* - 1 \quad \forall (\alpha,\beta) \in P(i,j);$$

2.4. Se $v \in V_R^s$ fazer: $\tilde{f}_{i,j}^s \leftarrow \tilde{f}_{i,j}^s + 1$

Caso contrário, fazer: $\tilde{f}_{i,j} \leftarrow \tilde{f}_{i,j} + 1;$

2.5. Se ainda existirem variáveis não nulas associadas a v , na solução ótima f^* , **voltar a 2.1.;**

Caso contrário, ir para 3.;

3. Se existirem vértices de V com variáveis associadas não nulas:

Considerar um qualquer, seja $v \in V;$

Voltar a 2.1.;

Caso contrário, Fim.

A solução \tilde{f} assim construída é, obviamente, admissível para o problema (B1').

Falta então provar que o valor associado a esta solução admissível, $z_{B1'}$, é igual a z_{B1}^* .

Cada unidade adicionada na variável $\tilde{f}_{i,j}^s$ foi determinada a partir de um caminho em vazio, de i para j , que intersecta V_R^s , ou seja, a partir de $P(i,j) = (i, \dots, u, v, \dots, j)$, com $v \in V_R^s$.

O custo unitário associado a $\tilde{f}_{i,j}^s$ é $l_{i,j}^s = \text{Min}_{v \in V_R^s} \{spl(i,v) + spl(v,j)\}$.

O custo do caminho $P(i,j)$ é igual a: $\sum_{(\alpha,\beta) \in P(i,j)} d_{\alpha,\beta}$.

Logo, $l_{i,j}^s$, representando o caminho de menor custo, em vazio, de i para j que intersecta V_R^s , verifica: $l_{i,j}^s \leq \sum_{(\alpha,\beta) \in P(i,j)} d_{\alpha,\beta}$.

Por outro lado, $P(i,j)$ é um caminho em vazio de i para j que intersecta V_R^s , na solução óptima de (B1), logo tem que ser o caminho de custo mínimo. Caso contrário podia ser substituído por um de custo total menor (o de custo mínimo) e a actual solução f^* não seria óptima.

Assim, $l_{i,j}^s = \sum_{(\alpha,\beta) \in P(i,j)} d_{\alpha,\beta}$.

De igual forma se prova que o custo unitário associado a cada variável $\tilde{f}_{i,j}$ de (B1') é igual ao custo do caminho que lhe deu origem, ou seja:

$$l_{i,j} = \sum_{(\alpha,\beta) \in P(i,j)} d_{\alpha,\beta}$$

Logo $z_{B1'} = z_{B1}^*$ e fica provado a).

b) Algoritmo: Construção de uma solução admissível para (B1) a partir de \tilde{f}^*

1. $\forall i, j \in V: f_{i,j} = 0$;

2. Se existir uma variável $\tilde{f}_{i,j}^{s*} > 0$ ainda não analisada,
ir para 3.

Caso contrário, seja $\tilde{f}_{i,j}^* > 0$ variável ainda não analisada;

Se existir, ir para 4.

Caso contrário, Fim.

3. Identificar o caminho de custo mínimo de i para j que intersecta V_R^s , seja:

$$P(i, j) = (i, \dots, u, v, \dots, j);$$

$$\text{Fazer: } f_{\alpha, \beta} = f_{\alpha, \beta} + \tilde{f}_{i, j}^{s*} \quad \forall (\alpha, \beta) \in P(i, j);$$

Voltar a 2.;

4. Identificar o caminho de custo mínimo de i para j , seja:

$$P(i, j) = (i, \dots, u, v, \dots, j);$$

$$\text{Fazer: } f_{\alpha, \beta} = f_{\alpha, \beta} + \tilde{f}_{i, j}^* \quad \forall (\alpha, \beta) \in P(i, j);$$

Voltar a 2..

A solução f assim construída é admissível para o problema (B1) e tem valor

$$z_{B1} = z_{B1}^*$$

pois, o custo do caminho de menor custo entre dois vértices é igual à soma dos custos dos arcos que fazem parte desse caminho.

Logo b) é verdadeira e o lema fica provado. \square

Ignorando em (B1') o conjunto de restrições (3.31), obtém-se a relaxação:

$$(B3) \quad \text{Min } z_{B3} = \sum_{j \in D} \sum_{i \in O} l_{i, j} \tilde{f}_{i, j} + \sum_{j \in D} \sum_{i \in O} l_{i, j}^s \tilde{f}_{i, j}^s \quad (3.29)$$

s.a :

$$\left\{ \begin{array}{l} \sum_{j \in D} \tilde{f}_{i, j} + \sum_{j \in D} \tilde{f}_{i, j}^s = a(i) \quad \forall i \in O \quad (3.30) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i \in O} \tilde{f}_{i, j} \leq b(j) \quad \forall j \in D \quad (3.32) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i \in O} \tilde{f}_{i, j}^s \leq b(j) \quad \forall j \in D \quad (3.33) \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{j \in D} \sum_{i \in O} \tilde{f}_{i, j}^s \geq K_s \quad (3.34) \end{array} \right.$$

$$\left\{ \begin{array}{l} \tilde{f}_{i, j} \geq 0 \text{ e inteiros} \quad \forall i \in O; \forall j \in D \quad (3.35) \end{array} \right.$$

$$\left\{ \begin{array}{l} \tilde{f}_{i, j}^s \geq 0 \text{ e inteiros} \quad \forall i \in O; \forall j \in D. \quad (3.36) \end{array} \right.$$

Este problema pode ser resolvido como um problema de fluxo de custo mínimo numa nova rede.

Considere-se então o grafo $\tilde{G} = (\tilde{V}, \tilde{A})$, onde $\tilde{V} = O \cup D \cup D^s \cup \{F, Y, T\}$, sendo:

- $D^s = \{\tilde{j}: j \in D\}$ o conjunto de réplicas dos destinos;
- F a origem da rede;
- T o destino da rede;
- Y um vértice utilizado para garantir (3.34), ou seja, que o fluxo que passa na componente conexa equilibrada tem o valor pretendido, igual a K_s .

Assim, todos os destinos duplicados são ligados a Y com um arco de custo nulo e de capacidade igual à procura no destino. É criado o arco (Y, T) de custo nulo e com capacidade igual ao valor que se fixa como limite inferior para o fluxo de (Y, T) , ou seja, igual a K_s .

Os arcos e os valores a estes associados, ou seja, o custo, a capacidade e o limite inferior ao valor do fluxo são definidos na tabela seguinte.

Arco $(i, j) \in \tilde{A}$	Custo $\tilde{\ell}_{i,j}$	Capacidade	Limite Inferior
$(F, i) \in \tilde{A}, \forall i \in O$	$\tilde{\ell}_{F,i} = 0$	$a(i)$	$a(i)$
$(i, j) \in \tilde{A}$ se $i \in O \wedge j \in D$	$\tilde{\ell}_{i,j} = \ell_{i,j}$	$\text{Min}\{a(i); b(j)\}$	0
$(i, \tilde{j}) \in \tilde{A}$ se $i \in O \wedge \tilde{j} \in D^s$	$\tilde{\ell}_{i,\tilde{j}} = \ell_{i,j}^s$	$\text{Min}\{a(i); b(j)\}$	0
$(\tilde{j}, Y) \in \tilde{A}$ se $\tilde{j} \in D^s$	$\tilde{\ell}_{\tilde{j},Y} = 0$	$b(j)$	0
$(Y, T) \in \tilde{A}$	$\tilde{\ell}_{Y,T} = 0$	K_s	K_s
$(j, T) \in \tilde{A}, \forall j \in D$	$\tilde{\ell}_{j,T} = 0$	$b(j)$	0

Tabela 3.4: Rede para o problema de fluxo de custo mínimo

Obviamente que resolver (B3) equivale a determinar o fluxo de custo mínimo de F para T na rede identificada pelo grafo \tilde{G} e pelos parâmetros do quadro 3.4. Designe-se este novo problema por $(\tilde{B3})$, de valor óptimo $z_{\tilde{B3}}^*$.

Proposição 3.3: $z_3 = C_{A_R} + z_{\tilde{B3}}^*$ é um minorante para o valor óptimo do PRRS, ou seja:

$$z_3 \leq z^*.$$

Prova: $z^* \geq z_{B1}^* + C_{A_R} = z_{B1'}^* + C_{A_R} \geq z_{B3}^* + C_{A_R} = z_{\tilde{B3}}^* + C_{A_R} \Rightarrow$
 $\Rightarrow z_3 = C_{A_R} + z_{\tilde{B3}}^* \leq z^*.$ □

Exemplo A: Para o exemplo que se tem considerado o grafo \tilde{G} é:

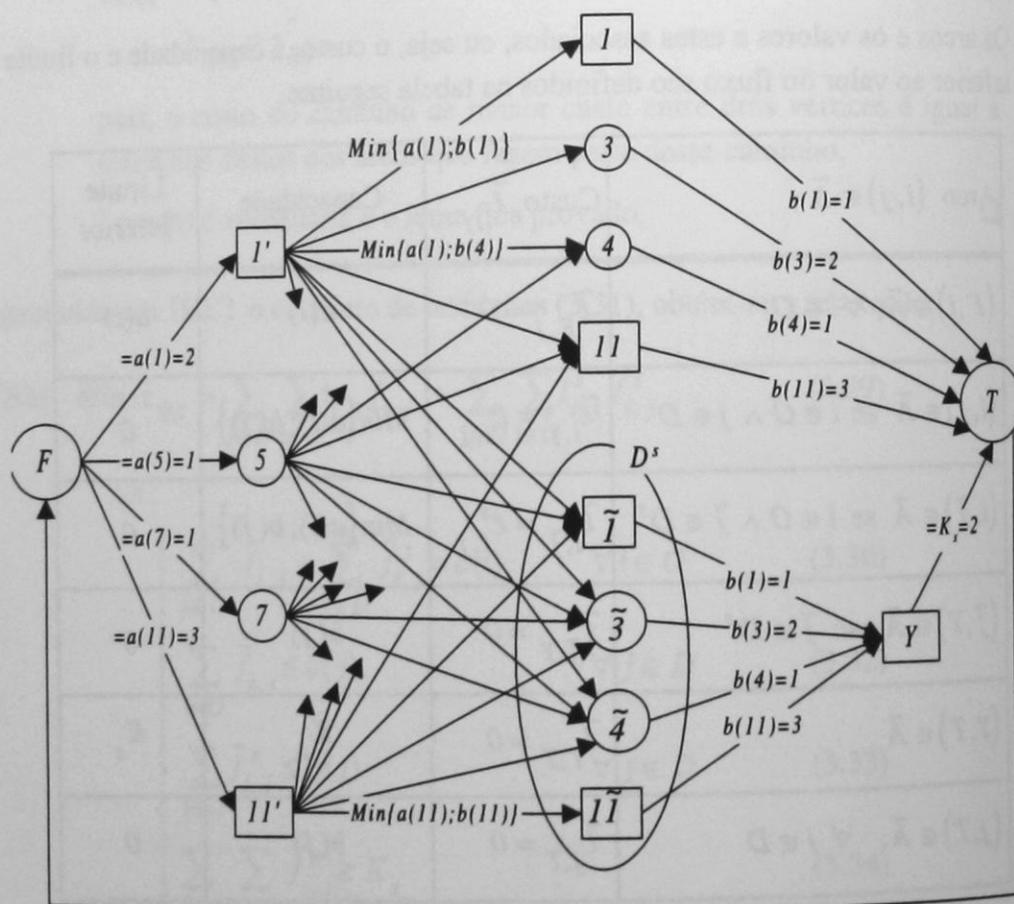


Figura 3.15: Rede associada ao problema de fluxo de custo mínimo

Com o objectivo de não sobrecarregar a figura, representaram-se apenas alguns dos arcos e as respectivas capacidades. Estas, quando antecedidas pelo sinal de igualdade indicam a existência de um limite inferior de igual valor.

Note-se que em qualquer solução óptima do problema de fluxo se tem sempre:

$$\sum_{i \in O} \tilde{f}_{F,i} = \sum_{i \in O} a(i) = \sum_{j \in D} b(j) = K_s + \sum_{j \in D} \tilde{f}_{j,T},$$

sendo neste exemplo a oferta total igual a 7.

Considere-se ainda que se determinou a seguinte solução óptima deste problema de fluxo:

$$\tilde{f}_{1',3} = 2; \quad \tilde{f}_{7,1} = 1; \quad \tilde{f}_{11',4} = 1; \quad \tilde{f}_{11',11} = 1; \quad \tilde{f}_{5,11}^s = 1; \quad \tilde{f}_{11',4}^s = 1.$$

Seja $(5, 8, 9, 11)$ o caminho de custo mínimo de 5 para 11 que intersecta V_R^s ;

$(11, 8, 5, 4)$ o caminho de custo mínimo de 11 para 4 que intersecta V_R^s .

Então, o multigrafo associado a esta solução é:

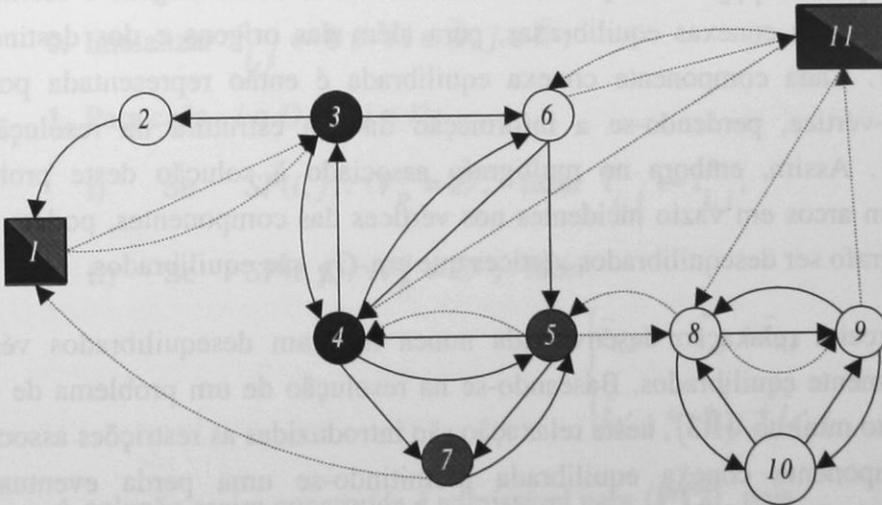


Figura 3.16: Multigrafo associado à solução de $(\tilde{\mathbf{B}}3)$

Neste caso o equilíbrio dos vértices de V_R^s é conseguido à custa da violação da procura nos destinos 4 e 11. \blacklozenge

Esta relaxação é sem dúvida a de resolução mais difícil, não só por se basear num problema de fluxo de custo mínimo em vez de num de transporte, como também por ser necessário considerar redes de maiores dimensões que nas relaxações anteriores. Por outro lado, só é possível ter em consideração uma das componentes conexas, não sendo incluídas restrições às restantes componentes que eventualmente existam.

3.5. *Comparações entre os Minorantes*

Neste ponto vão estabelecer-se algumas propriedades para comparar os valores óptimos das relaxações. Embora se prove que nenhuma das relaxações domina as restantes, estas comparações vão ser possíveis em determinadas instâncias.

Relembre-se então que na primeira relaxação é resolvido um problema de transporte, (PT1), com o objectivo de equilibrar todos os vértices que no grafo de procuras estão desequilibrados impondo, em simultâneo, grau mínimo nos dois vértices especiais, l e $n+1$. O valor óptimo de (PT1) foi designado por z_{PT1}^* .

A segunda relaxação é também baseada num problema de transporte, (PT2), de valor óptimo z_{PT2}^* . Neste problema consideram-se como origens e destinos as componentes conexas equilibradas, para além das origens e dos destinos de (PT1). Cada componente conexa equilibrada é então representada por um super-vértice, perdendo-se a informação da sua estrutura na resolução de (PT2). Assim, embora no multigrafo associado à solução deste problema existam arcos em vazio incidentes nos vértices das componentes, podem neste multigrafo ser desequilibrados vértices que em G_R são equilibrados.

Na terceira relaxação desenvolvida nunca resultam desequilibrados vértices inicialmente equilibrados. Baseando-se na resolução de um problema de fluxo de custo mínimo, ($\tilde{B}3$), nesta relaxação são introduzidas as restrições associadas à componente conexa equilibrada permitindo-se uma perda eventual do equilíbrio dos destinos de G_R , garantido nas duas relaxações anteriores. O valor óptimo deste problema é $z_{\tilde{B}3}^*$.

No que se segue, e dado na terceira relaxação só serem incluídas restrições referentes a uma componente conexa, supõe-se $nc = 1$. Simplifica-se assim a exposição sendo os resultados referentes às duas primeiras relaxações generalizáveis para o caso de existirem duas ou mais componentes conexas equilibradas.

Considere-se então que G_R é formado por pelo menos duas componentes conexas, sendo apenas uma delas equilibrada. Represente-se por V_R^s o conjunto de vértices na componente conexa equilibrada, e por s o super-vértice associado.

Seja K_s definido, como habitualmente, por (3.21).

Instâncias em que a primeira relaxação domina a segunda podem então ser identificadas com base na proposição seguinte.

Proposição 3.4: Se à solução óptima de (PT1) estiverem associados pelo menos K_s caminhos de custo mínimo, de origens de O para destinos de D , que incluam vértices de V_R^s , então

$$z_{PT1}^* \geq z_{PT2}^*.$$

Prova: Seja $t_{i,j}^*$, $i \in O, j \in D$ uma solução óptima de (PT1).

Se a $t_{i,j}^*$, $i \in O, j \in D$ estiverem associados exactamente ⁽¹⁾ K_s caminhos de custo mínimo, de origens de O para destinos de D , que intersectem V_R^s , esta solução pode ser transformada numa solução admissível de (PT2), como se segue:

0. Inicializar $\hat{t}_{i,j} \leftarrow 0; \forall i \in \hat{O}, j \in \hat{D}$;

1. Para cada $i \in O$ e $j \in D$:

i) Se $SP(i,j) \cap V_R^s = \emptyset$, fazer $\hat{t}_{i,j} \leftarrow t_{i,j}^*$;

ii) Se $SP(i,j) \cap V_R^s \neq \emptyset$, fazer:

$$\begin{cases} \hat{t}_{i,s} \leftarrow t_{i,j}^* + \hat{t}_{i,s} \\ \hat{t}_{s',j} \leftarrow t_{i,j}^* + \hat{t}_{s',j} \end{cases}$$

A solução assim construída é admissível para (PT2), pois

• Se $i \in O$

$$\sum_{j \in \hat{D}} \hat{t}_{i,j} = \sum_{\substack{j \in D \\ SP(i,j) \cap V_R^s = \emptyset}} t_{i,j}^* + \sum_{\substack{j \in D \\ SP(i,j) \cap V_R^s \neq \emptyset}} t_{i,j}^* = a(i)$$

• Se $i = s$

$$\sum_{j \in \hat{D}} \hat{t}_{s',j} = K_s$$

(1) Se existirem mais de K_s caminhos nas condições do enunciado transformar a solução óptima de (PT1) numa solução admissível de (PT2) pode ser feito com base no algoritmo que se apresenta, repetindo 1.ii) para os primeiros K_s caminhos e 1.i) para os restantes.

$$\therefore \sum_{j \in \hat{D}} \hat{l}_{i,j} = \hat{a}(i) \quad \forall i \in \hat{O}.$$

Prova-se de forma semelhante que $\sum_{i \in \hat{O}} \hat{l}_{i,j} = \hat{b}(j) \quad \forall j \in \hat{D}$.

Logo, a solução construída é admissível para (PT2).

Seja z_{PT2} o valor desta solução admissível. Cada um de K_s caminhos de custo mínimo de $i \in O$ para $j \in D$ foi substituído por dois caminhos de custo mínimo: $SP(i, s)$ e $SP(s', j)$.

Nestes casos, $l_{i,j} = l_{i,u} + l_{u,v} + l_{v,j}$ com $u, v \in V_R^s$.

Logo,

$$\begin{aligned} l_{i,j} &= l_{i,u} + l_{u,v} + l_{v,j} \geq l_{i,u} + l_{v,j} \stackrel{(3.28)}{\geq} \hat{l}_{i,s} + \hat{l}_{s',j} \Rightarrow \\ &\Rightarrow z_{PT1}^* \geq z_{PT2} \geq z_{PT2}^*. \end{aligned}$$

□

Identificam-se de seguida instâncias em que a desigualdade da proposição é estrita.

Corolário 3.1: Se à solução óptima de (PT1) estiverem associados pelo menos K_s caminhos de custo mínimo, de origens de O para destinos de D , que intersectem V_R^s e pelo menos um dos arcos de um desses caminhos tem os dois extremos em vértices de V_R^s , então

$$z_{PT1}^* > z_{PT2}^*.$$

Prova: pela proposição 3.4, nas condições do corolário, $z_{PT1}^* \geq z_{PT2}^*$.

Seja $i \in O$, $j \in D$ e $SP(i, j) = SP(i, u) \cup (u, v) \cup SP(v, j)$, $u, v \in V_R^s$.

Então,

$$\begin{aligned} l_{i,j} &= l_{i,u} + l_{u,v} + l_{v,j} > l_{i,u} + l_{v,j} \stackrel{(3.28)}{\geq} \hat{l}_{i,s} + \hat{l}_{s',j} \Rightarrow \\ &\Rightarrow z_{PT1}^* > z_{PT2} \geq z_{PT2}^*. \end{aligned}$$

□

Identificadas instâncias em que a primeira relaxação domina a segunda, identificam-se de seguida instâncias em que a terceira relaxação é dominada também pela primeira.

Proposição 3.5: Se à solução óptima de **(PT1)** estiverem associados pelo menos K_s caminhos de custo mínimo, de origens de O para destinos de D , que intersectem V_R^s , então

$$z_{PT1}^* \geq z_{B3}^*.$$

Prova: Seja $t_{i,j}^*$, $i \in O, j \in D$ uma solução óptima de **(PT1)**.

Se a $t_{i,j}^*$, $i \in O, j \in D$ estiverem associados pelo menos K_s caminhos de custo mínimo de origens de O para destinos de D que intersectem V_R^s , esta solução pode ser transformada numa solução admissível de **(B3)**, como se segue:

0. Fazer $\tilde{f}_{i,j} \leftarrow 0; \tilde{f}_{i,j}^s \leftarrow 0; \forall i \in \tilde{O}, j \in \tilde{D};$

1. Para cada $i \in O$ e $j \in D$

i) Se $SP(i,j) \cap V_R^s = \emptyset$, fazer $\tilde{f}_{i,j} \leftarrow t_{i,j}^*$;

ii) Se $SP(i,j) \cap V_R^s \neq \emptyset$, fazer $\tilde{f}_{i,j}^s \leftarrow t_{i,j}^*$.

A solução assim construída é admissível para **(B3)**, pois verifica as restrições:

• (3.30), se $i \in O$

$$\sum_{j \in D} (\tilde{f}_{i,j} + \tilde{f}_{i,j}^s) = \sum_{\substack{j \in D \\ SP(i,j) \cap V_R^s = \emptyset}} t_{i,j}^* + \sum_{\substack{j \in D \\ SP(i,j) \cap V_R^s \neq \emptyset}} t_{i,j}^* = a(i).$$

• (3.32), se $j \in D$

$$\sum_{i \in O} \tilde{f}_{i,j} = \sum_{\substack{i \in O \\ SP(i,j) \cap V_R^s = \emptyset}} t_{i,j}^* \leq b(j).$$

• (3.33), se $j \in D$

$$\sum_{i \in O} \tilde{f}_{i,j}^s = \sum_{\substack{i \in O \\ SP(i,j) \cap V_R^s \neq \emptyset}} t_{i,j}^* \leq b(j).$$

- (3.34), existindo pelo menos K_s caminhos a intersectar V_R^s ,

$$\sum_{j \in D} \sum_{i \in O} \tilde{f}_{i,j}^s \geq K_s.$$

O valor desta solução admissível, z_{B3} , é igual a z_{PT1}^* , logo,

$$z_{PT1}^* = z_{B3} \geq z_{B3}^*.$$

□

Nas proposições anteriores foram identificadas algumas das instâncias em que a primeira relaxação domina as duas restantes. De seguida vão estabelecer-se condições para dominância das duas últimas em relação à primeira. Fica assim provado, como já se tinha sugerido, que nenhuma das relaxações domina outra.

Proposição 3.6: Se o multigrafo associado à solução óptima de (PT2) é equilibrado, então

$$z_{PT2}^* \geq z_{PT1}^*.$$

Prova: Seja $\hat{t}_{i,j}^*$, $i \in \hat{O} \wedge j \in \hat{D}$ uma solução óptima de (PT2), nas condições da proposição.

Cada caminho de $i \in O$ para $j \in D$ que intersecta V_R^s pode decompor-se em dois caminhos, um caminho de i para um vértice $v \in V_R^s$ e outro de $v \in V_R^s$ para j .

Logo, sob as hipóteses da proposição, $\sum_{i \in O} \hat{t}_{i,v}^* = \sum_{j \in D} \hat{t}_{v,j}^*$, $\forall v \in V_R^s$.

Construa-se a solução $t_{i,j}$, $i \in O, j \in D$ como se segue:

0. Inicializar $t_{i,j} \leftarrow 0$; $\forall i \in O, j \in D$;

1. Para cada par $i \in O, j \in D$:

Se $(\hat{t}_{i,s}^* > 0$ e $\hat{t}_{s',j}^* > 0)$

fazer: $t_{i,j} \leftarrow \text{Min} \{ \hat{t}_{i,s}^* ; \hat{t}_{s',j}^* \}$;

$\hat{t}_{i,s}^* \leftarrow \hat{t}_{i,s}^* - t_{i,j}$;

$\hat{t}_{s',j}^* \leftarrow \hat{t}_{s',j}^* - t_{i,j}$;

$t_{i,j} \leftarrow t_{i,j} + \hat{t}_{i,j}^*$;

Caso contrário, fazer: $t_{i,j} \leftarrow \hat{t}_{i,j}^*$.

A solução $t_{i,j}$, $i \in O$, $j \in D$ assim construída é solução admissível de (PT1), pois:

$$\begin{cases} \sum_{j \in D} t_{i,j} = \sum_{j \in \hat{D}} \hat{t}_{i,j} = a(i) \quad \forall i \in O \\ \sum_{i \in O} t_{i,j} = \sum_{i \in \hat{O}} \hat{t}_{i,j} = b(j) \quad \forall j \in D. \end{cases}$$

Por outro lado, na construção da solução admissível de (PT1), de valor z_{PT1} , caminhos de menor custo de i para j que intersectavam V_R^s foram substituídos por caminhos de menor custo de i para j . Logo,

$$z_{PT2}^* \geq z_{PT1} \geq z_{PT1}^*. \quad \square$$

Foram então identificadas instâncias em que a primeira relaxação domina a segunda e vice-versa. Exemplifica-se de seguida que se na solução óptima de (PT1) não existirem pelo menos K_s caminhos de menor custo a intersectar V_R^s e se a solução óptima de (PT2) não proporcionar um multigrafo equilibrado, então as duas relaxações não são comparáveis.

Exemplo 3.5: Considere-se que os arcos representados na figura seguinte são a única diferença entre as soluções óptimas de (PT1) e de (PT2).

Assim, $t_{i,j}^* = 1$ na solução óptima de (PT1). Na solução óptima de (PT2), $\hat{t}_{i,s}^* = 1$ e $\hat{t}_{s',j}^* = 1$.

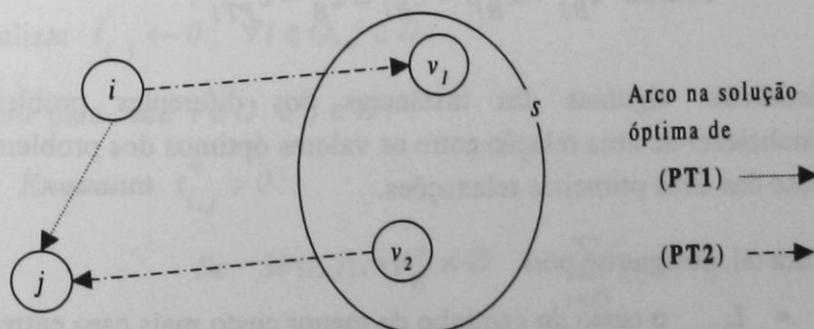


Figura 3.17: Diferenças entre as soluções de (PT1) e (PT2)

Em (PT1) o arco (i,j) representando o caminho de custo mínimo conduz a

$$l_{i,j} \leq l_{i,v_1} + l_{v_1,v_2} + l_{v_2,j}.$$

Logo

$$\text{se } \ell_{i,j} > \ell_{i,v_1} + \ell_{v_2,j} \Rightarrow z_{PT1}^* > z_{PT2}^* ;$$

$$\text{se } \ell_{i,j} < \ell_{i,v_1} + \ell_{v_2,j} \Rightarrow z_{PT1}^* < z_{PT2}^* ;$$

$$\text{se } \ell_{i,j} = \ell_{i,v_1} + \ell_{v_2,j} \Rightarrow z_{PT1}^* = z_{PT2}^* .$$

Instâncias em que a terceira relaxação domina a primeira são identificadas de seguida.

Proposição 3.7: Seja \tilde{F}^* uma solução óptima de **(B3)**.

$$\text{Se } \forall j \in D, \sum_{i \in O} \tilde{f}_{i,j}^* + \sum_{i \in O} \tilde{f}_{i,j}^{s*} = b(j)$$

$$\text{então, } z_{B3}^* \geq z_{PT1}^* .$$

Prova: Se $\forall j \in D, \sum_{i \in O} \tilde{f}_{i,j}^* + \sum_{i \in O} \tilde{f}_{i,j}^{s*} = b(j)$

então as restrições relaxadas de **(B1')** são verificadas.

$$\text{Logo, } z_{B3}^* = z_{B1'}^* .$$

$$\text{Por outro lado, pelo lema 3.5, } z_{B1'}^* = z_{B1}^* .$$

$$\text{Assim, } z_{B3}^* = z_{B1'}^* = z_{B1}^* \geq z_B^* = z_{PT1}^* .$$

Estudadas algumas das instâncias dos diferentes problemas vai ainda estabelecer-se uma relação entre os valores óptimos dos problemas de transporte base das duas primeiras relaxações.

Para tal, designe-se por:

- L_{\max} o custo do caminho de menor custo mais caro entre dois vértices de V_R^s ;
- $scl(s)$ o custo do circuito de custo mínimo que contém s na rede associada a **(B2)**.

Proposição 3.8: $z_{PT2}^* - K_s scl(s) \leq z_{PT1}^* \leq z_{PT2}^* + K_s L_{\max}$.

Prova:

(a) Prove-se primeiro a desigualdade: $z_{PT2}^* - K_s scl(s) \leq z_{PT1}^*$.

Seja $\delta \geq 0$ o número de caminhos de custo mínimo (de origens de O para destinos de D) que intersectam V_R^s , associados à solução ótima de (PT1).

Seja $t_{i,j}^*$, $i \in O, j \in D$ uma solução ótima de (PT1).

Para construir uma solução admissível de (PT2) a partir da solução ótima de (PT1), há que considerar $\delta \leq K_s$ ou $\delta > K_s$.

1º caso. $\delta \leq K_s$

A solução:

$$\begin{cases} \hat{t}_{i,j} = t_{i,j}^* & \forall i \in O, j \in D \\ \hat{t}_{s',s} = K_s \end{cases}$$

é uma solução admissível para (PT2) de valor

$$z_{PT2} = z_{PT1}^* + K_s scl(s) \geq z_{PT2}^*.$$

2º caso. $\delta > K_s$

A solução ótima de (PT1) pode ser transformada numa solução admissível de (PT2), como se segue:

0. Inicializar $\hat{t}_{i,j} \leftarrow 0$; $\forall i \in \hat{O}, j \in \hat{D}$;

1. *Repetir* para cada $i \in O$ e $j \in D$:

Enquanto $t_{i,j}^* > 0$:

Se $SP(i,j) \cap V_R^s \neq \emptyset$ e $\sum_{i \in O} \hat{t}_{i,s} < K_s$

fazer: $\hat{t}_{i,s} \leftarrow \hat{t}_{i,s} + 1$;

$\hat{t}_{s',j} \leftarrow \hat{t}_{s',j} + 1$;

$t_{i,j}^* \leftarrow t_{i,j}^* - 1$

Caso contrário, fazer: $\hat{t}_{i,j} \leftarrow \hat{t}_{i,j} + t_{i,j}^*$;
 $t_{i,j}^* \leftarrow 0$.

Esta solução admissível tem valor

$$z_{PT2} \leq z_{PT1}^*$$

pois para todas as variáveis $t_{i,j}^* > 0$ para as quais $SP(i,j) \cap V_R^s \neq \emptyset$ existem vértices $v_1, v_2 \in V_R^s$ tais que

$$SP(i,j) = SP(i,v_1) \cup SP(v_1,v_2) \cup SP(v_2,j),$$

e portanto

$$\begin{aligned} \ell_{i,j} &= spl(i,j) = spl(i,v_1) + spl(v_1,v_2) + spl(v_2,j) \geq \\ &\geq spl(i,v_1) + spl(v_2,j) \geq \\ &\geq \hat{\ell}_{i,s} + \hat{\ell}_{s',j} = \hat{\ell}_{i,j}. \end{aligned}$$

Logo, $z_{PT2}^* \leq z_{PT2} \leq z_{PT1}^* \leq z_{PT1} + K_s scl(s)$.

(b) Falta provar que: $z_{PT1}^* \leq z_{PT2}^* + K_s L_{\max}$

Numa solução óptima de (PT2) tem-se

$$\sum_{i \in O} \hat{t}_{i,s}^* \leq K_s \quad \text{e} \quad \sum_{j \in D} \hat{t}_{s',j}^* \leq K_s.$$

Para cada par $(i,s), (s',j)$, representativos dos arcos iniciais (i,v_1) e (v_2,j) , considere-se o $SP(v_1,v_2)$.

Logo, $SP(i,v_1) \cup SP(v_1,v_2) \cup SP(v_2,j) = P(i,j)$.

Por outro lado, $z_{PT2}^* + \sum_{(v_1,v_2)} spl(v_1,v_2) \leq z_{PT2}^* + K_s L_{\max}$.

A solução óptima de (PT2) pode ser transformada numa solução admissível de (PT1), como se segue:

0. Inicializar $t_{i,j} \leftarrow \hat{t}_{i,j}^*; \forall i \in O, j \in D$;

1. *Repetir* para cada $i \in O$: $\hat{t}_{i,s}^* > 0$

Encontrar $j \in D$: $\hat{t}_{s',j}^* > 0$:

Fazer: $w = \text{Min}\{\hat{t}_{s',j}^*; \hat{t}_{i,s}^*\}$

$$t_{i,j} \leftarrow t_{i,j} + w;$$

$$\hat{t}_{i,s}^* \leftarrow \hat{t}_{i,s}^* - w;$$

$$\hat{t}_{s',j}^* \leftarrow \hat{t}_{s',j}^* - w.$$

Assim,

$$z_{PT1}^* \leq z_{PT1} \leq z_{PT2}^* + \sum_{(v_1, v_2)} \text{spl}(v_1, v_2) \leq z_{PT2}^* + K_s L_{\max}. \quad \square$$

O termo heurística vem do termo grego *heuriskō* que significa descobrir ou encontrar. Reeves (26) frisa que muitas vezes este termo é usado para métodos exatos, em que se pretende "encontrar" a solução ótima dos problemas. No contexto desta tese, tal como usualmente em investigação operacional, este termo é usado para métodos em que se determinam "boas" soluções admissíveis (quase ótimas) num tempo de execução aceitável, mas relacionados com uma "procura" que propiamente vem uma "descoberta".

Reeves (26) define o seu livro sobre heurísticas a

... those who share ideas without the necessity of infallible certainties and enjoy the wonder of confusion.

1.1. Introdução

As heurísticas são uma ferramenta importante na determinação de soluções admissíveis para problemas de difícil resolução. Sendo de minimização o objectivo do PRRS, o valor de tais soluções representa um limite inferior para o seu valor óptimo. Pretende-se assim a criação de heurísticas que forneçam soluções admissíveis de valores de relaxação quanto possível do valor óptimo.

Handwritten text, possibly a date or reference number.

Handwritten text, possibly a name or title.

Handwritten text, possibly a list or notes.

Handwritten text, possibly a name or title.

Handwritten text, possibly a list or notes.

Capítulo 4. Heurísticas para o PRRS

O termo heurística vem do termo grego *heuriskein* que significa descobrir ou encontrar. Reeves [80] frisa que muitas vezes este termo é usado para métodos exactos, em que se pretende “encontrar” a solução óptima dos problemas. No contexto desta tese, tal como usualmente em investigação operacional, este termo é usado para métodos em que se determinam “boas” soluções admissíveis (*quase óptimas*) num tempo de execução aceitável, mais relacionados com uma “procura” que propriamente com uma “descoberta”.

Forstic ⁽¹⁾ dedica o seu livro sobre heurísticas a

“those who share ideas without the necessity of infallible certitudes and enjoy the wonder of confusion”.

4.1. Introdução

As heurísticas são uma ferramenta importante na determinação de soluções admissíveis para problemas de difícil resolução. Sendo de minimização o objectivo do PRRS, o valor de tais soluções representa um majorante para o seu valor óptimo. Põe-se assim a questão de desenvolver heurísticas que forneçam soluções admissíveis de valores tão próximos quanto possível do valor óptimo.

⁽¹⁾ Forstic, G., 1987, *Heuristic*, 1st. ed. Presscraft Press, Benzonia, Michigam.

Uma solução admissível para o PRRS é formada por tantas rotas quantos os veículos. Relembre-se que cada rota é um conjunto de subrotas admissíveis formado por uma subrota inicial (de 1 para $n+1$), uma final (de $n+1$ para 1) e, eventualmente, por subrotas intermédias (de $n+1$ para $n+1$), que não excedam a capacidade dos veículos.

A geração de um conjunto de subrotas admissíveis pode considerar-se simples num (multi)grafo Euleriano. O multigrafo solução do método LB, apresentado no capítulo anterior, associado à primeira relaxação, é formado por componentes conexas Eulerianas. Como anteriormente referido, em geral, este multigrafo não constitui uma solução admissível para o PRRS, pois podendo ser formado por diversas componentes conexas, pode não ser possível, em cada componente, obter subrotas admissíveis compatíveis com a capacidade dos veículos. Assim, as heurísticas construtivas que se apresentam têm como objectivo a transformação da solução de LB, em geral não admissível, numa solução admissível, recorrendo a propriedades conhecidas de grafos Eulerianos.

Uma descrição global das heurísticas é feita em 4.2. que, genericamente, podem ser consideradas heurísticas construtivas de três fases: I) geração de subrotas admissíveis; II) agregação de subrotas; III) obtenção de uma solução admissível, com tantas subrotas iniciais e finais quantos os veículos. Como se verá, o desenvolvimento de métodos alternativos nas duas primeiras fases origina três heurísticas diferentes.

Nos pontos seguintes são então apresentados e exemplificados métodos para cada uma destas fases, sendo para as duas primeiras fases apresentados métodos alternativos. Assim, na geração de subrotas admissíveis é apresentado um método que se baseia na partição de um "circuito gigante" e outro que tem como objectivo a determinação de subrotas com procura total baixa, se comparada com a capacidade dos veículos.

A agregação é feita ou de uma forma sequencial, como se descreve em 4.4., ou tendo em conta a proximidade das subrotas (ponto 4.5.). A junção por proximidade é ainda subdividida em dois processos diferentes, um que se baseia na determinação de emparelhamentos e outro na identificação de sementes.

Como se verá, em 4.6., a construção de uma solução admissível a partir de um conjunto de subrotas admissíveis é quase imediata.

Em 4.7. é feito um estudo teórico aos valores obtidos por uma das heurísticas, sendo estabelecido um majorante para o pior desvio do valor da heurística, relativamente ao valor do minorante.

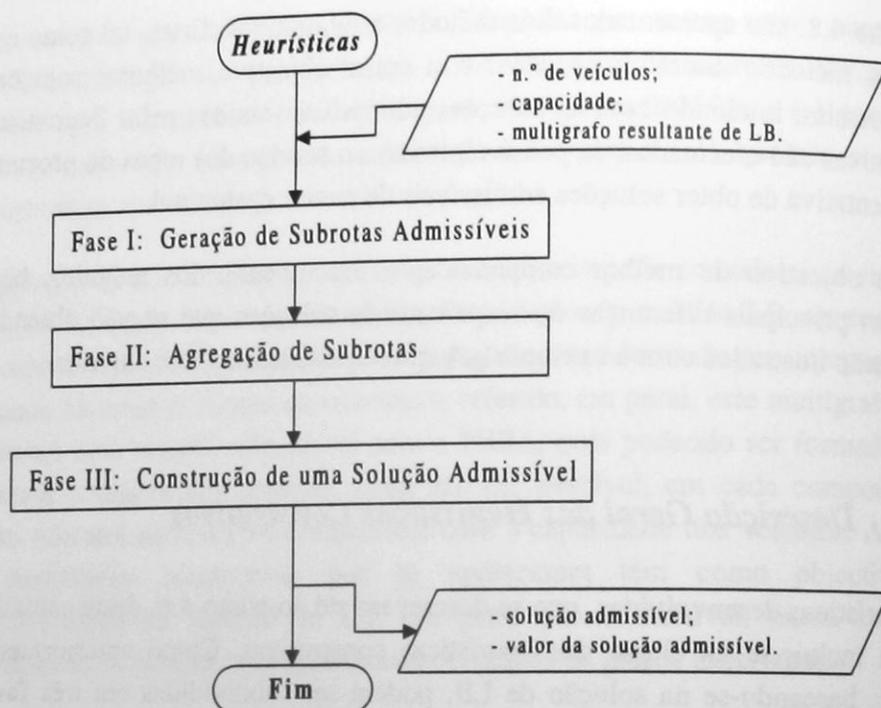
No ponto 4.8. são apresentados dois métodos melhorativos. Estes, tal como em geral os métodos assim designados, têm como objectivo melhorar soluções dadas. Assim, iniciando com as soluções admissíveis obtidas pelas heurísticas construtivas são efectuadas, se possível, trocas ao serviço dos arcos de procura, numa tentativa de obter soluções admissíveis de menor custo total.

Com o objectivo de melhor compreender o desempenho dos métodos, bem como as principais diferenças no respeitante às soluções que se vão obtendo, estes serão ilustrados com o exemplo **A** apresentado no capítulo anterior.

4.2. Descrição Geral das Heurísticas Construtivas

As heurísticas desenvolvidas, que se descrevem até ao ponto 4.6. deste capítulo, podem incluir-se na classe das heurísticas construtivas. Como anteriormente frisado, baseando-se na solução de LB, podem ser subdivididas em três fases principais (fluxograma 4.1).

Na primeira fase são construídas subrotas admissíveis, a partir do multigrafo solução do método LB. Esta tarefa vem simplificada dado se estar em presença de um multigrafo formado por componentes conexas Eulerianas. A junção de subrotas, quando possível, origina, em geral, o corte de ligações em vazio e, conseqüentemente, a obtenção de um conjunto de subrotas mais favorável, em termos do valor da solução. Foi então criada a segunda fase onde se tentam agregar algumas das subrotas geradas na primeira fase. Na terceira fase é criada uma solução admissível a partir do conjunto de subrotas em presença, pois, um conjunto de subrotas admissíveis só proporciona uma solução admissível caso possa ser subdividido num conjunto de P rotas. Num conjunto de P rotas, lembre-se, têm que existir P subrotas iniciais, P subrotas finais, sendo as restantes subrotas intermédias. Nesta fase é então analisada a solução e altera-se o vértice inicial e/ou o vértice final de alguma(s) subrota(s), se necessário.



Fluxograma 4.1: Heurísticas

Num (multi)grafo Euleriano é possível identificar, sem muito esforço computacional, um circuito (ou um conjunto de circuitos) que passe uma e uma só vez por cada arco [45]. Como é óbvio, circuitos nestas condições facilitam a geração de subrotas admissíveis, isto é, subrotas intermédias, iniciais ou finais, compatíveis com a capacidade dos veículos. Mostra-se então vantajosa a utilização de grafos Eulerianos na geração de subrotas admissíveis.

O método LB baseia-se no cálculo do custo de serviço e na resolução de um problema de transporte (PT1). Neste (PT1), as origens e os destinos são definidos a partir dos vértices desequilibrados, ou seja, os vértices em que os graus internos e externos diferem, no grafo de procura, e da imposição de certos valores para os graus internos e externos dos vértices 1 e $n+1$. Os valores assumidos pelas variáveis de decisão na solução óptima de (PT1), $t_{i,j}^*$, representam, como se viu,

- o número de vezes que devem ser copiados os caminhos de menor custo em vazio de i (ou da sua representação na rede inicial se $i = 1', (n+1)'$) para j , se i e j não representam o mesmo vértice na rede inicial, para que todos os vértices resultem equilibrados;
- o número de vezes que deve ser copiado o circuito de custo mínimo em vazio que inclui j , se i e j representam o mesmo vértice no grafo inicial, ou seja se $(i = 1' e j = 1)$ ou $(i = (n+1)' e j = n+1)$, para que 1 e $n+1$ tenham como graus interno e externo certos valores estabelecidos.

Seja:

A_{PTI} a lista de arcos na solução óptima de (PT1), em que cada arco (i, j) , ou a sua representação na rede inicial, é repetido $t_{i,j}^*$ vezes;

$G_{LB} = (V, A_{LB})$ o multigrafo solução do método LB, onde $A_{LB} = A_R \cup A_{PTI}$ ⁽¹⁾.

Euler ⁽²⁾ provou que todos os vértices de um grafo orientado serem equilibrados é condição necessária e suficiente para que esse grafo seja Euleriano. Logo, todas as componentes conexas de G_{LB} representam sub-multigrafos Eulerianos.

Tendo como objectivo a transformação da solução de LB numa solução admissível, nas heurísticas que se descrevem começa por gerar-se subrotas admissíveis no multigrafo G_{LB} .

São apresentados dois métodos para a construção das subrotas admissíveis em G_{LB} , denominados por: *método de geração por partição de um circuito Euleriano* – GSCE; *método de geração por procuras* – GSProc.

Em GSCE identifica-se em primeiro lugar um circuito “gigante” que será depois subdividido num conjunto de subrotas admissíveis. O principal objectivo do método GSProc é o da construção de diversas subrotas “pequenas”, isto é, subrotas em que a procura total é “pequena” quando comparada com a capacidade dos veículos, W .

Relembre-se que cada arco (i, j) resultante da resolução de (PT1) representa o caminho de custo mínimo em vazio de i para j no grafo inicial, G . Em qualquer dos métodos de geração de subrotas optou-se por manter esta representação. Assim, no decorrer destes métodos, um arco que é atravessado em vazio representa sempre o caminho de menor custo em vazio, em G , entre os seus respectivos extremos, tendo associado o custo desse caminho, $spl(i, j)$. Tal opção facilita a implementação computacional e permite a obtenção de rotinas mais eficientes.

(1) $A_{LB} = A_R \cup A_{PTI}$ representa também uma lista de arcos repetidos.

(2) Veja-se Ford e Fulkerson [39].

Esta mesma representação foi ainda considerada em qualquer das outras fases das heurísticas. Como se justificará, embora possam não ser contempladas algumas hipóteses de solução, pensa-se ser computacionalmente compensador o recurso a tal representação.

Na fase de agregação de subrotas foram também criados dois métodos: *método de agregação sequencial* – AgSeq; *método de agregação por proximidade* – AgProxim. Em AgSeq as subrotas são agregadas em função da ordem por que foram geradas. No método AgProxim as subrotas são agregadas se forem consideradas adjacentes. AgProxim é ainda subdividido em dois métodos alternativos: *método de emparelhamentos* – AgE e *método de sementes* – AgS, como se descreverá.

A terceira fase consiste apenas num teste simples à admissibilidade da solução, como se descreverá no algoritmo SolAdm. Como frisado, é nesta altura que se testa a existência de exactamente P subrotas iniciais e de P subrotas finais. Caso o número de subrotas deste tipo não coincida com o número de veículos, P , altera-se o início e/ou o fim de alguma(s) subrota(s).

O valor da solução admissível, majorante para o valor óptimo do PRRS, é dado por:

$$\bar{z} = C_{A_R} + \sum spl(i, j),$$

onde (i, j) representam os arcos nos caminhos percorridos em vazio na solução admissível gerada.

Assim, embora se esteja sempre perante um processo construtivo com três fases, consoante o método usado em cada fase obtêm-se distintas soluções admissíveis, logo diferentes heurísticas. Como se justificará não faz sentido considerar todas as possibilidades de conjugação destes métodos, o que poderia levar à obtenção de seis heurísticas. Em suma, com estes métodos podem obter-se três heurísticas distintas, como se esquematiza na tabela seguinte.

Heurística	Fase I	Fase II	Fase III
H1	GSCE	AgSeq	SolAdm
H2	GSProc	AgE	
H3	GSProc	AgS	

Tabela 4.1: Sequência de métodos nas diferentes heurísticas

4.3. Geração de Subrotas

4.3.1. Introdução

A geração de subrotas é feita de duas formas alternativas. Em primeiro lugar, e tendo-se determinado um multigrafo formado por componentes Eulerianas, identifica-se um circuito Euleriano, em cada componente, que será subdividido num conjunto de subrotas admissíveis, ou seja, subrotas iniciais, finais e intermédias, compatíveis com a capacidade dos veículos. Este método, de geração por partição, é descrito no ponto seguinte.

Em problemas de determinação de rotas para veículos com restrições de capacidade e procura nos arcos e/ou vértices, mostra-se, por vezes, muito penalizadora a subdivisão de circuitos “gigantes”, como o circuito Euleriano. Pois, tais partições, considerando apenas a admissibilidade da solução em função da capacidade dos veículos, esquecendo a localização do depósito, podem originar a introdução de ligações ao depósito de pontos muito afastados deste. Intuitivamente, para um depósito “afastado” dos restantes vértices do grafo, este facto mostrar-se-á tanto mais penalizador quanto menos ligações existirem no grafo inicial ao depósito e quanto mais afastados estiverem os restantes vértices do grafo entre si.

No PRRS, $n + 1$ encontra-se significativamente afastado dos restantes vértices, ou seja, da zona de recolha, não se considerando, porém, indiferente a escolha dos pontos em que se devem interromper as subrotas, como propõem Álvarez-Valdés et al. [1]. Havendo “poucos” vértices do grafo ligados a $n + 1$, na rede inicial, faz sentido pensar que estes são os melhores pontos de interrupção de subrotas.

Uma vantagem deste método consiste em originar uma heurística para a qual é possível estabelecer, como se verá, um majorante para o erro relativo, medido em função do desvio do valor da solução admissível relativamente ao do minorante calculado pelo algoritmo LB. Assim, pode estabelecer-se *a priori*, para cada instância do PRRS, um majorante para o pior erro relativo que se obterá.

Idealmente, as subrotas teriam início num vértice de procura “próximo” de $n + 1$, ir-se-iam desenvolvendo de forma a que quando a capacidade dos veículos exigisse a sua interrupção, se estivesse novamente num vértice “próximo” de $n + 1$, não deixando por servir arcos muito afastados das ligações

a $n+1$. Não sendo trivial a geração de subrotas que satisfaçam este tipo de requisitos, nos métodos de procura que se sugerem nesta tese pretende gerar-se subrotas que, ao invés de serem subdivididas, possam vir a ser agregadas. Assim, tenta-se começar com a construção de diversas subrotas "pequenas" (representando subrotas em que a procura total é "baixa"), que serão posteriormente agregadas.

Os dois métodos de geração propostos são ilustrados em simultâneo no final da exposição, em 4.3.4., com o exemplo protótipo **A**.

4.3.2. Método de Geração por Partição de um Circuito Euleriano

A ideia subjacente ao método de geração por partição é muito simples, do estilo de heurísticas anteriormente sugeridas para problemas similares por Beltrami e Bodin [6], Bodin e Kursh [11], [12], Frederickson et al. [41], Newton e Thomas [68], Stern e Dror [87] e por Ulusoy [90].

Sendo o multigrafo solução do método para determinar minorantes, G_{LB} , formado por diversos subgrafos Eulerianos, torna-se possível gerar circuitos gigantes em G_{LB} identificando circuitos Eulerianos neste multigrafo. Estes circuitos podem ser subdivididos num conjunto de subrotas admissíveis. Trata-se, como se definiu anteriormente, de subrotas iniciais (de 1 para $n+1$), finais (de $n+1$ para 1), ou intermédias (de $n+1$ para $n+1$), que não excedam a capacidade dos veículos, como se descreve no algoritmo GSCE.

As subrotas vão sendo criadas de acordo com a sequência de arcos do circuito Euleriano. Uma subrota é interrompida sempre que, ao seguir esta sequência, for encontrado um dos vértices especiais (1 ou $n+1$), ou a inclusão do seguinte arco de procura na subrota não seja compatível com a capacidade dos veículos. Neste caso, a subrota é terminada com uma ligação a $n+1$, sendo iniciada outra subrota com uma ligação inicial de $n+1$ ao vértice inicial do arco de procura em análise.

Uma subrota com início em 1 , representando uma subrota inicial, tem que terminar em $n+1$. Logo, na geração de subrotas na componente que contém 1 há que impedir que termine em 1 uma subrota com início em 1 . Obviamente que as subrotas finais terão como extremo inicial $n+1$ e como extremo final 1 .

Se 1 e $n+1$ não pertencem à componente conexa em foco, o primeiro arco que se considera na subrota é um arco de $n+1$ ao vértice inicial do primeiro arco de procura na sequência de arcos visitados pelo circuito Euleriano dessa

componente. Obviamente que no final, após terem sido considerados todos os arcos de procura da componente, há que considerar um arco do último vértice incluído a $n+1$. Pois, o início e o fim de qualquer subrota tem que coincidir ou com o vértice 1 ou com $n+1$.

Algoritmo GSCE: Geração de Subrotas Admissíveis por Partição de Circuitos Eulerianos

0. Seja $G_{LB} = (V, A_{LB})$ o multigrafo associado à solução de LB;
 nc o número de componentes conexas em G_{LB} ;
 $ms = |A_R^s|$; n.º de arcos de procura na s -ésima componente conexa;

Fazer: $s \leftarrow 1$; { s representa o número da componente conexa}

1. Construir um circuito gigante, identificando um circuito Euleriano, CE , na componente conexa s , com início em $n+1$, caso este vértice pertença à componente conexa, ou com início no vértice de índice mais baixo, no caso contrário;

Seja $CE = (a_1, a_2, \dots, a_k, \dots, a_{mce})$, com $a_k = (i_k, j_k)$;

Se $i_1 \neq 1, n+1$ Iniciar a nova subrota: $\{(n+1, i_1); a_1\}$;

C.c., Iniciar a nova subrota: $\{a_1\}$;

Fazer: $k \leftarrow 2$; { k é o número do arco em CE }

2. Se $i_k = 1$

Se a subrota actual tem início em 1

Retirar a_{k-1} da subrota;

Terminar a subrota com o arco $(i_{k-1}, n+1)$;

Se $a_{k-1} = (i_{k-1}, 1)$ é percorrido em serviço

Criar uma subrota final $\{(n+1, i_{k-1}), a_{k-1}\}$;

C.c., Terminar a subrota final actual;

Iniciar a nova subrota: $\{a_k\}$;

C.c., Se $i_k = n+1$ Terminar a subrota actual;

Iniciar a nova subrota: $\{a_k\}$;

C.c., *Se* a inclusão de a_k na subrota não é compatível com a capacidade dos veículos

Terminar a subrota, com uma ligação a $n+1$;

Iniciar uma nova subrota $\{(n+1, i_k), a_k\}$;

C.c., Incluir a_k na subrota;

3. Fazer $k \leftarrow k+1$ ⁽¹⁾

Se Os m_s arcos de procura já foram incluídos em alguma subrota

Terminar a subrota, com uma ligação a $n+1$;

Ir para 4.

C.c., *Voltar a* 2.;

4. Fazer $s \leftarrow s+1$

Se $s \leq nc$ *Voltar a* 1.

C.c., *Fim.*

A geração do(s) circuito(s) Euleriano(s) foi feita com base no algoritmo apresentado em *pseudo-Pascal* no livro de Gondran e Minoux [45]. Considerou-se que o vértice $n+1$ é o vértice inicial do circuito Euleriano na componente conexa que o contém. Para as restantes componentes conexas os circuitos Eulerianos têm início no vértice de índice mais baixo da respectiva componente.

Cada arco nas subrotas geradas, ou representa um arco com procura positiva, ou representa o caminho de menor custo em vazio entre os respectivos vértices.

No passo 2. de GSCE, sempre que uma subrota é interrompida por ter sido encontrado o vértice l e caso não se trate de uma subrota com início em l , essa subrota é considerada como subrota final, ou seja, termina em l . Se a subrota em questão tiver início em l , terá que ter $n+1$ como último vértice. É sempre necessário ter em consideração que subrotas admissíveis são de l para $n+1$, de $n+1$ para $n+1$ ou de $n+1$ para l .

Note-se ainda que quando se interrompe uma subrota por ter sido encontrado l ou $n+1$ podem ser geradas subrotas em vazio, ou seja, subrotas em que não existem arcos de procura positiva. Não sendo necessárias, tais subrotas são sempre eliminadas, isto é, nunca são consideradas para a fase seguinte.

⁽¹⁾ Na implementação computacional incrementa-se k até a_k ser um arco de procura.

O aspecto mais negativo deste método está no forçar de ligações a $n+1$, estando este vértice significativamente afastado dos vértices do grafo de procuras. Sendo o circuito Euleriano determinado sem ter em consideração custos e procuras totais incluídas no circuito, pode mostrar-se penalizador ter que considerar tais ligações, mesmo sabendo que o multigrafo de base a este circuito foi otimizado no respeitante ao custo dos arcos em vazio incluídos. Perante várias alternativas possíveis, o único factor que influencia a escolha do próximo arco a considerar no circuito Euleriano, é o facto de existirem arcos de procura positiva e de procura nula a incidir num mesmo vértice. Ou seja, sempre que existem diferentes arcos com o mesmo vértice inicial, consideram-se em primeiro lugar os arcos de procura positiva para formarem o circuito Euleriano. Tenta-se assim, não chegar a precisar de todos os arcos de procura nula quando da partição deste circuito em subrotas admissíveis.

Possivelmente este método poderá vir a ser melhorado se for alterada a actual identificação do circuito Euleriano, de forma a que se seja considerada tanto a procura total no circuito como o custo de eventuais ligações a $n+1$. Assim, poderia, por exemplo, ser considerado um arco que mais se afaste de $n+1$, caso a procura total no circuito seja baixa, em relação à capacidade dos veículos, sendo, no caso contrário, escolhido um arco que mais se aproxime de $n+1$.

Existindo, num mesmo grafo, vários circuitos Eulerianos, poder-se-ia ainda pensar se seria vantajoso testar a qualidade de solução face a diferentes circuitos Eulerianos iniciais. Porém, pensa-se que dispor de diferentes circuitos não vai melhorar significativamente e de forma compensadora, face ao esforço computacional adicional exigido, a qualidade da solução.

Este tipo de métodos, de partição de um circuito "gigante", conduzem, em geral, à geração de subrotas desequilibradas, sem que possam vir a ser agregadas, na sua maioria. Na realidade verifica-se que nas primeiras subrotas geradas em cada componente conexa, a quantidade total de resíduos será aproximadamente igual à capacidade dos veículos, enquanto nas últimas subrotas geradas para cada componente, a quantidade de resíduos a recolher será, tendencialmente, baixa, quando comparada com a capacidade dos veículos. Como se verá nos resultados computacionais, na maior parte dos casos são poucas as subrotas passíveis de ser agregadas, quando geradas desta forma, resultando no final subrotas algo desequilibradas.

4.3.3. Método de Geração por Procuras

Alternativamente à geração de subrotas por partição de um circuito gigante, pode pensar-se em métodos que iniciem com um conjunto de "pequenas" subrotas. Estas podem posteriormente ser agregadas com vista à obtenção de soluções admissíveis de menor valor. O primeiro método deste tipo foi apresentado, em 1964, por Clarke e Wright [23], para problemas com procura nos vértices.

Em certas heurísticas propostas sob esta perspectiva considera-se que cada arco de procura dá origem, inicialmente, a uma subrota distinta ([42] e [43]). Outro tipo de circuitos iniciais foram sugeridos em [7], [30], [44], [48], [62] e [76].

Tendo então como objectivo a construção de circuitos "pequenos" (no respeitante à procura total incluída em cada circuito), começa por identificar-se, em G_{LB} , o circuito de procura mínima que contém cada vértice. A(s) primeira(s) subrota(s) são construídas a partir do circuito com maior procura total, entre os circuitos identificados. Retiram-se do grafo os arcos incluídos neste circuito e repete-se o processo enquanto existirem arcos de procura no grafo. Tenta-se assim gerar circuitos que possam, mais tarde, vir a ser agregados, dado a procura total em cada um ser baixa, se comparada com a capacidade dos veículos. O algoritmo GSProc esquematiza o método.

Neste método não são considerados, caso existam, os arcos de G_{LB} correspondentes ao circuito de procura mínima que contém $n+1$, pois tais arcos nunca dão origem a circuitos com procura positiva e podem inibir a detecção de circuitos de procura que incluam os restantes vértices do grafo.

Algoritmo GSProc: Geração de Subrotas Admissíveis por Identificação de Circuitos de Procura Mínima

0. Seja $G_{LB} = (V, A_{LB})$ o multigrafo resultante do método LB;

Fazer: $G'' = (V'', A'')$, com $V'' = V$ e $A'' = A_{LB} \setminus \{SC(n+1)\}$ ⁽¹⁾,

⁽¹⁾ Retirando de A_{LB} todas as réplicas de arcos no circuito de custo mínimo que contém $n+1$, ou seja, em $SC(n+1)$.

1. Em G'' identificar o circuito de procura mínima que contém cada vértice (algoritmo de Floyd);

2. Seleccionar o circuito com maior procura total, entre os circuitos identificados em 1.;

Sejam V_C e A_C o conjunto de vértices e o conjunto de arcos no circuito seleccionado, respectivamente;

3. Construir subrotas admissíveis a partir do circuito seleccionado em 2..

Se $l \in V_C$

Tendo em conta W , construir uma subrota inicial e/ou uma subrota final e, eventualmente, subrotas intermédias, a partir dos arcos de A_C , começando com o arco que tem início em l ;

C.c., Construir, a partir dos arcos de A_C , subrotas admissíveis e intermédias, iniciando pelo vértice de V_C a menor custo de $n + l$;

4. Retirar de G'' os arcos no circuito seleccionado em 2. e os vértices em que já não incidam arcos em A'' ;

5. *Se* $A'' \cap A_R = \emptyset$ **Fim**

C.c., **Voltar a 1.**

Caminhos mais curtos entre qualquer par de vértices num grafo podem ser identificados pelo algoritmo de Floyd [38] ⁽¹⁾. Se a matriz utilizada for a das procuras, estes caminhos representarão, obviamente, caminhos de procura mínima entre qualquer par de vértices. Logo, circuitos de procura mínima contendo cada vértice podem ser determinados com base no algoritmo de Floyd aplicado à matriz das procuras, em que se considera inicialmente infinita a procura na diagonal principal. Assim, estes valores, no final do algoritmo, representam a procura nos circuitos de procura mínima que contém cada vértice. O valor mais elevado desta diagonal é seleccionado, de forma a tentar evitar deixar para o fim circuitos maiores, no respeitante à procura. A identificação dos arcos no circuito seleccionado é feita a partir da matriz auxiliar dos caminhos que vai sendo actualizada no decorrer das iterações do algoritmo de Floyd.

⁽¹⁾ Nos programas desenvolvidos no âmbito desta tese foi utilizada a implementação em Pascal apresentada em [88].

Inicialmente, na matriz de procuras, utilizada no algoritmo de Floyd, considera-se a procura em cada arco coincidente com a sua procura na rede inicial. Assim, cada arco de procura que apareça replicado no multigrafo, é em primeiro lugar considerado como arco de procura positiva. Em cada iteração do algoritmo a matriz de procuras é actualizada sendo retirados os arcos no circuito seleccionado. Nesta actualização há que saber sempre se ainda existem réplicas de cada um dos arcos por considerar uma vez que são as réplicas, obtidas com a resolução do problema de transporte, que garantem a existência dos circuitos que contêm qualquer vértice em qualquer iteração do método, como se prova de seguida.

Proposição 4.1: Em qualquer passo do algoritmo, qualquer vértice ou tem grau interno e externo nulos ou existe em G'' um circuito que o contém.

Prova: Inicialmente, $G'' = (V'', A'') = G_{LB}$.

Então, G'' é formado por componentes conexas Eulerianas, em que

$$g^-(v) = g^+(v), \quad \forall v \in V''.$$

Em cada passo do algoritmo, é retirado de G'' um circuito que contém um vértice. Logo, o grafo G'' resultante, em cada passo, é formado por componentes conexas Eulerianas, ou seja, em cada passo do algoritmo,

$$g^-(v) = g^+(v), \quad \forall v \in V''.$$

Logo, existe sempre um circuito que contém cada vértice de V'' , ou o vértice foi retirado de V'' , no passo 4., por ter grau zero em G'' . \square

A construção de subrotas a partir de circuitos que não incluam o vértice l é sempre iniciada pelo vértice do circuito a menor custo de $n+1$. Se l pertencer ao circuito e $n+1$ não, são construídas subrotas a partir de l , resultando sempre, pelo menos, uma subrota inicial ou uma subrota final. Assim, se a procura total no circuito, Q_C , for tal que

$$\triangleright Q_C \leq W \text{ e}$$

- apenas um dos arcos incidentes em l é servido, gera-se ou uma subrota inicial ou uma subrota final, tendo em atenção esse arco.
- os dois arcos incidentes em l são de procura nula gera-se uma subrota inicial.
- os dois arcos incidentes em l são de procura positiva gera-se uma subrota inicial e uma final.

$$\triangleright W < Q_C \leq 2W$$

é construída uma subrota inicial e uma subrota final. Para tal inicia-se a subrota inicial no arco que tem l como vértice inicial e esta subrota é interrompida quando a capacidade dos veículos o exigir. A partir da interrupção da subrota inicial é então construída a subrota final.

$$\triangleright Q_C > 2W$$

é construída pelo menos uma subrota intermédia, uma inicial e uma final.

Em suma, em cada iteração de GSProc, o circuito seleccionado pode dar origem a mais de uma subrota. Isto verifica-se no caso em que um dos vértices especiais, l ou $n+l$, está incluído no circuito, ou se a procura total excede a capacidade dos veículos. Espera-se, porém, que os circuitos sejam pequenos no respeitante à procura total neles incluída, não sendo expectável que cada circuito dê origem a diversas subrotas.

Tal como no método de geração anterior, a construção das subrotas nunca considera subrotas em vazio. Porém, neste método, as subrotas são formadas a partir de circuitos que têm sempre procura positiva, dada a forma como são criados. Assim, só há que evitar a geração de subrotas em vazio, quando é necessário subdividir o circuito de partida.

4.3.4. Exemplo

Exemplo A: Considere-se de novo o exemplo **A** introduzido no capítulo anterior, referente a um problema com 2 veículos de capacidades $W = 4$, e o multigrafo resultante do método para determinar minorantes representado na figura 3.9 da página 54.

Seguidamente exemplificam-se os dois métodos descritos para a geração de subrotas admissíveis. Como se verá, são obtidas subrotas distintas.

Algoritmo GSCE

0. Em G_{LB} existem $nc = 2$ componentes conexas; $m1 = 13$; $m2 = 6$
1. Construir um circuito gigante na primeira componente conexa, identificando um circuito Euleriano, CE .

Na figura 4.1 os números sobre os arcos indicam a sequência dos arcos visitados pelo CE identificado.

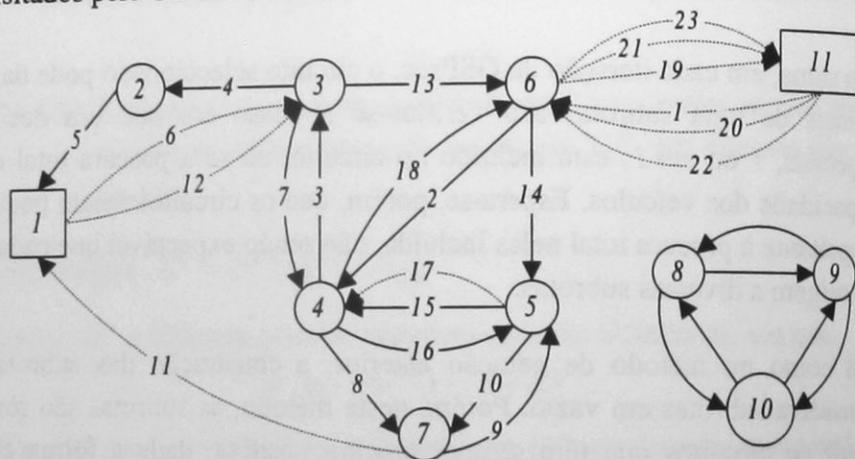


Figura 4.1: CE para a primeira componente conexa de G_{LB}

2. e 3. Geração de subrotas admissíveis a partir de CE

Seguindo a sequência de arcos de CE são geradas, na primeira componente conexa, as subrotas:

$$S_1 = \{(11,6); (6,4); (4,3); (3,2); (2,1)\} \quad \text{com } Q_1 = 4 = W$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\} \quad \text{com } Q_2 = 4 = W$$

$$S_3 = \{(11,3); (3,6); (6,5); (5,4); (4,5); (5,11)\} \quad \text{com } Q_3 = 4 = W$$

$$S_4 = \{(11,4); (4,6); (6,11)\} \quad \text{com } Q_4 = 1 < W.$$

1. Identificar um circuito Euleriano na segunda componente conexa. Tal como anteriormente, na figura 4.2 os números sobre os arcos indicam a sequência dos arcos visitados pelo CE identificado, com início no vértice de menor índice.

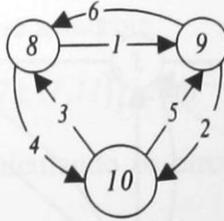


Figura 4.2: CE para a segunda componente conexa de G_{LB}

2. e 3. A geração de subrotas admissíveis a partir de CE, iniciando pelo vértice mais próximo de $n + 1 = 11$, conduz a

$$S_5 = \{(11,8); (8,9); (9,10); (10,8); (8,10); (10,11)\} \text{ com } Q_5 = 4$$

$$S_6 = \{(11,10); (10,9); (9,8); (8,11)\} \text{ com } Q_6 = 2.$$

Foram assim geradas, por GSCE, seis subrotas admissíveis, sendo uma delas inicial e outra final. Tal como tinha sido frisado neste conjunto de subrotas apenas duas subrotas têm procura total pequena, relativamente a W . São as últimas subrotas geradas em cada componente conexa, S_4 com procura 1 e S_6 com procura 2. Assim, este conjunto de subrotas será semelhante ao conjunto final de subrotas, dado a capacidade dos veículos não permitir alterações substanciais.

Algoritmo GSProc

0. $G'' = (V'', A'') = (V, A_{LB} \setminus \{SC(n+1)\});$

1. Em G'' identificar o circuito de procura mínima que contém cada vértice. Na figura 4.3 os valores perto dos vértices representam a procura dos circuitos de procura mínima que contém cada vértice. Cada arco replicado é sempre considerado em primeiro lugar como arco de procura, como é o caso de (5,4).

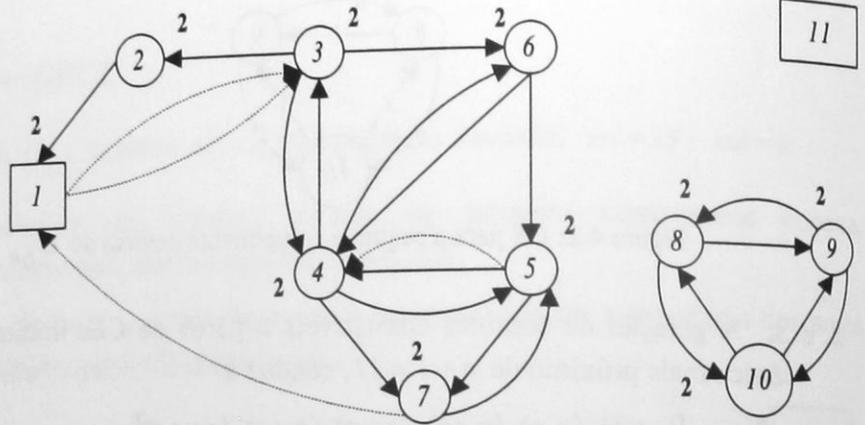


Figura 4.3: Procura nos circuitos de procura mínima que contêm cada vértice de G_{LB}

2. Selecciona-se o circuito com procura total máxima, igual a 2, que contêm o vértice 1, onde:

$$V_C = \{1, 2, 3\}; \quad A_C = \{(1,3); (3,2); (2,1)\}.$$

3. A partir deste circuito gera-se a primeira subrota admissível

$$S_1 = \{(11,3); (3,2); (2,1)\} \quad \text{com} \quad Q_1 = 2.$$

Sendo o arco $(2,1)$ um arco de procura positiva, o arco $(1,3)$ um arco de procura nula e a procura total no circuito inferior a $W = 4$, é gerada uma subrota final.

4. e 1. A actualização de G'' , e o cálculo dos circuitos de procura mínima no novo grafo conduz a:

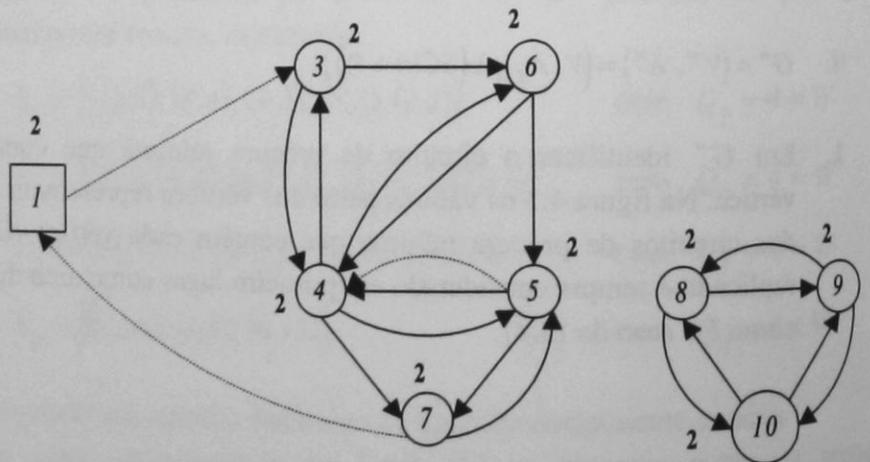


Figura 4.4: Procura nos circuitos de procura mínima que contêm cada vértice de G''

2. Um circuito de procura total máxima tem procura igual a 2 e, por exemplo:

$$V_C = \{1, 3, 4, 7\}; \quad A_C = \{(1,3); (3,4); (4,7); (7,1)\}.$$

3. A partir deste circuito gera-se a subrota

$$S_2 = \{(1,3); (3,4); (4,7); (7,11)\} \text{ com } Q_2 = 2.$$

4.e 1. Actualizando G'' e calculando os circuitos de procura mínima no novo grafo obtém-se:

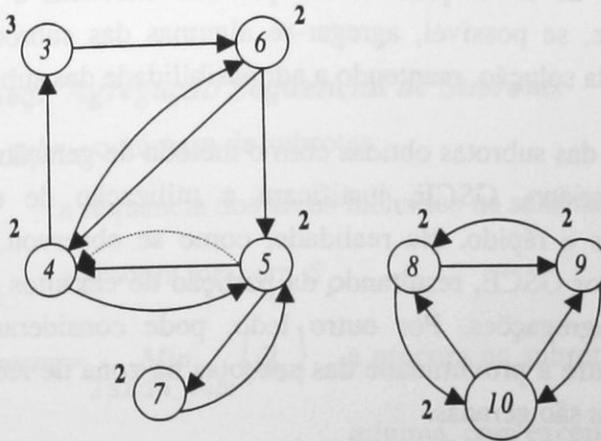


Figura 4.5: Procura nos circuitos de procura mínima que contêm cada vértice de G''

2. O circuito de procura total máxima tem procura igual a 3 e:

$$V_C = \{3, 4, 6\}; A_C = \{(3,6); (6,4); (4,3)\}.$$

3. A partir deste circuito gera-se a subrota:

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\} \text{ com } Q_3 = 3,$$

pois 6 é o vértice de V_C a menor custo de $n + 1 = 11$.

Prosseguindo desta forma, resultam ainda as subrotas:

$$S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\} \text{ com } Q_4 = 3$$

$$S_5 = \{(11,5); (5,7); (7,5); (5,11)\} \text{ com } Q_5 = 2$$

$$S_6 = \{(11,8); (8,9); (9,8); (8,11)\} \text{ com } Q_6 = 2$$

$$S_7 = \{(11,8); (8,10); (10,8); (8,11)\} \text{ com } Q_7 = 2$$

$$S_8 = \{(11,9); (9,10); (10,9); (9,11)\} \text{ com } Q_8 = 2$$

$$S_9 = \{(11,4); (4,5); (5,11)\} \text{ com } Q_9 = 1.$$

Foram neste caso geradas nove subrotas, com procura total, em geral, baixa. Ao contrário do que acontece no método anterior, as subrotas obtidas por GSProc podem, na sua maioria, vir a ser agregadas. ♦

4.4. Agregação Sequencial de Subrotas

Dado um conjunto de subrotas admissíveis, isto é, subrotas de l para a $n+1$, de $n+1$ para l , ou de $n+1$ para $n+1$, que não excedam a capacidade dos veículos, pretende, se possível, agregar-se algumas das subrotas, de forma a diminuir o valor da solução, mantendo a admissibilidade das subrotas.

As características das subrotas obtidas com o método de geração por partição de um circuito Euleriano, GSCE, justificam a utilização de um método de agregação simples e rápido. Na realidade, como se observou, o conjunto de subrotas gerado por GSCE, resultando da partição de circuitos Eulerianos, não permite muitas agregações. Por outro lado, pode considerar-se que existe alguma relação entre a proximidade das subrotas na zona de recolha e a ordem por que as mesmas são geradas.

No método de agregação sequencial são agregadas subrotas respeitando a ordem dos seus índices, ou seja, a ordem por que foram geradas. Começa-se por calcular $procm$ como sendo a procura na subrota de procura total mínima, entre as subrotas geradas. Respeitando a ordem por que as subrotas foram geradas identifica-se, então, a primeira subrota cuja procura total adicionada a $procm$ não exceda W , ou seja, a primeira subrota que poderá vir a ser eventualmente agregada. É então analisada a vantagem de juntar a subrota identificada com algumas das suas subsequentes.

Duas subrotas S_i e S_j só são agregadas se for vantajoso, i.e., se a subrota resultante, S_r , originar uma solução de menor custo total que a solução actual, escolhendo a melhor das junções, i.e., entre:

$$S_r \leftarrow S_i \cup S_j \text{ e } S_r \leftarrow S_j \cup S_i$$

escolhe-se a que origina a solução de menor custo. A agregação entre subrotas é neste método considerada apenas entre os seus extremos. Ou seja, $S_r \leftarrow S_i \cup S_j$ consiste em considerar em S_r os arcos pela seguinte ordem:

- os arcos de S_i até ao último arco que é servido nesta subrota;
- um arco de ligação entre o último arco servido em S_i e o primeiro servido em S_j ;
- a sequência seguinte dos arcos de S_j , a partir do que é servido em primeiro lugar.

A primeira subrota identificada só é "abandonada", passando-se a uma sua subsequente em condições de poder ser agregada, quando já não for possível agregá-la com nenhuma das subrotas suas subsequentes. O processo termina quando já não é possível agregar mais subrotas, como se resume no algoritmo AgSeq.

Algoritmo AgSeq: Agregação Sequencial de Subrotas

0. Seja: n_subr o número de subrotas;

S_k a sequência dos arcos incluídos na subrota k ;

Q_k a procura total em S_k ;

$procm = \underset{2 \leq k \leq n_subr}{Min} \{Q_k\}$ a procura na subrota de procura total mínima, com excepção da primeira;

Fazer: $k_1 \leftarrow 1$ {n.º da subrota que se vai tentar agregar com alguma das suas subsequentes};

1.

1.1 Se $Q_{k_1} + procm > W$ Ir para 1.4;

C.c., Fazer $k_2 \leftarrow k_1 + 1$;

1.2 Se $Q_{k_1} + Q_{k_2} > W$

ou (S_{k_1} é inicial e S_{k_2} tem um extremo em l)

ou (S_{k_1} é final e S_{k_2} tem um extremo l)

Ir para 1.3;

C.c.,

- Agregar as subrotas S_{k_1} e S_{k_2} se for vantajoso, designando por S_{k_1} a subrota agregada e retirando S_{k_2} do conjunto de subrotas;
- Actualizar as subrotas e as respectivas procuras totais;
- Fazer: $n_subr \leftarrow n_subr - 1$;
- Se $procm = Q_{k_2}$ Fazer: $procm = \underset{k_1+1 \leq k \leq n_subr}{Min} \{Q_k\}$;
- Voltar a 1.1;

1.3 Fazer $k_2 \leftarrow k_2 + 1$;

Se $k_2 \leq n_subr$ e $Q_{k_1} + procm < W$ Voltar a 1.2

1.4 Fazer $k_1 \leftarrow k_1 + 1$;

Se $k_1 < n_subr$

Se $procm = Q_{k_1}$ Fazer: $procm = \underset{k_1+1 \leq k \leq n_subr}{\text{Min}} \{Q_k\}$;

Voltar a 1.1;

C.c., Fim.

Em qualquer dos métodos de agregação que se apresentam admite-se a fusão de subrotas se a capacidade dos veículos o permitir e se pelo menos uma das subrotas for intermédia (ou seja, tiver início e termo em $n+1$). Assim, subrotas iniciais (finais) só serão, eventualmente, agregadas com subrotas intermédias. Tal foi fundamentalmente motivado pelo facto de que admitir outro tipo de agregações poderia significar admitir a agregação de subrotas que teriam, mais tarde, que ser desagregadas. Este ponto será justificado de seguida.

• **Agregar: Subrota Inicial – Subrota Final**

Para que esta agregação não origine uma subrota não admissível do vértice l para ele próprio, só faria sentido considerar a subrota agregada como uma subrota que passa em l . Ou seja, a subrota resultante da agregação da

subrota inicial $S_i = \{(l, i_1, i_2, \dots, i_\ell, n+1)\}$

com a subrota final $S_j = \{(n+1, j_1, j_2, \dots, j_k, l)\}$

seria $S_j \cup S_i = \{(n+1, j_1, j_2, \dots, j_k, l, i_1, i_2, \dots, i_\ell, n+1)\}$.

Sendo necessária a existência de um número de subrotas iniciais (finais) igual ao número de veículos, poder-se-ia estar, nesta fase, a agregar subrotas que teriam de ser desagregadas na fase III da heurística. Note-se ainda que este tipo de agregações não origina a eliminação de ligações a $n+1$, que são as de maior custo. Por outro lado, a junção de uma subrota intermédia com uma subrota inicial (final) faz com que seja eliminada uma ligação a $n+1$. Assim, as subrotas iniciais (finais) só se agregam, inicialmente, com subrotas intermédias. As agregações deste tipo, ou seja, entre subrotas iniciais e finais, são consideradas apenas na terceira fase das heurísticas, caso possível e vantajoso.

• **Agregar: Subrota Inicial – Subrota Inicial**

Considerem-se as duas subrotas iniciais:

$$S_i = \left\{ \left(l, i_1, i_2, \dots, i_\ell, n+1 \right) \right\} \quad \text{e} \quad S_j = \left\{ \left(l, j_1, j_2, \dots, j_k, n+1 \right) \right\}.$$

A subrota resultante da agregação de S_i com S_j seria ou

$$S_i \cup S_j = \left\{ \left(l, i_1, i_2, \dots, i_\ell, l, j_1, j_2, \dots, j_k, n+1 \right) \right\}$$

ou

$$S_j \cup S_i = \left\{ \left(l, j_1, j_2, \dots, j_k, l, i_1, i_2, \dots, i_\ell, n+1 \right) \right\},$$

ou seja, sempre uma subrota com início em l . Assim, de duas subrotas iniciais, obtinha-se apenas uma inicial. Tendo em conta o que se disse no caso anterior, optou-se por admitir este tipo de agregação apenas no final, se possível.

Estas considerações mantêm-se válidas para a agregação de duas subrotas finais.

Note-se que o objectivo deste método é o de, se possível, agregar subrotas de uma forma simples e, conseqüentemente, rápida. Logo, não faz sentido considerar a possibilidade de junção de subrotas sem ser entre os seus extremos. Ou seja, uma subrota é sempre incluída no fim ou no início de outra e nunca a meio, mesmo que estas tenham vértices em comum. Assim, sendo possível agregar as subrotas intermédias:

$$S_i = \left\{ \left(n+1, i_1, i_2, \dots, i_\ell, n+1 \right) \right\} \quad \text{e} \quad S_j = \left\{ \left(n+1, j_1, j_2, \dots, j_k, n+1 \right) \right\},$$

escolhe-se para subrota agregada, a que proporcionar uma solução de menor valor, entre as duas seguintes:

$$S_i \cup S_j = \left\{ \left(n+1, j_1, j_2, \dots, j_k, i_1, i_2, \dots, i_\ell, n+1 \right) \right\};$$

$$S_j \cup S_i = \left\{ \left(n+1, i_1, i_2, \dots, i_\ell, j_1, j_2, \dots, j_k, n+1 \right) \right\}.$$

O aspecto mais negativo deste método está no facto de não ser considerada nenhuma medida que meça a proximidade das subrotas na zona de recolha, nem o afastamento destas relativamente a $n+1$. Assim, tendo sido desenvolvido para ser aplicado sobre o conjunto de subrotas obtido a partir do método de geração por partição, GSCE, mostra-se, como seria de esperar, pouco eficaz para o conjunto de subrotas gerado a partir de GSProc. Tal provém fundamentalmente de em GSCE as subrotas que podem eventualmente ser agregadas nesta fase serem poucas. Contrariamente, em GSProc deverão existir

diversas subrotas passíveis de ser agregadas, sendo que só ocasionalmente poderá existir alguma relação entre os índices das subrotas, ordem por que foram geradas, e a sua proximidade.

Por sua vez, estando $n+1$ afastado dos restantes vértices do grafo, mostra-se sempre vantajosa a fusão de duas quaisquer subrotas, podendo este método originar agregações longe de representarem as melhores fusões entre as subrotas, como se mostra de seguida com o exemplo **A**.

Assim, o método de agregação sequencial, AgSeq, tendo surgido por ser simples, rápido e proporcionar soluções "razoáveis", se forem consideradas as subrotas geradas por GSCE, deve ser abandonado se for considerado outro conjunto inicial de subrotas, nomeadamente o obtido com GSProc. Tal motivou o desenvolvimento dos métodos de agregação por proximidade, que, na medida do possível, têm em conta a proximidade das subrotas quer na zona de recolha, quer relativamente a $n+1$.

Exemplo A: considerem-se as subrotas geradas pelo método GSCE para o exemplo **A**, no qual se assumiram disponíveis 2 veículos de capacidade $W = 4$.

$$S_1 = \{(11,6); (6,4); (4,3); (3,2); (2,1)\} \quad \text{com } Q_1 = 4$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\} \quad \text{com } Q_2 = 4$$

$$S_3 = \{(11,3); (3,6); (6,5); (5,4); (4,5); (5,11)\} \quad \text{com } Q_3 = 4$$

$$S_4 = \{(11,4); (4,6); (6,11)\} \quad \text{com } Q_4 = 1$$

$$S_5 = \{(11,8); (8,9); (9,10); (10,8); (8,10); (10,11)\} \quad \text{com } Q_5 = 4$$

$$S_6 = \{(11,10); (10,9); (9,8); (8,11)\} \quad \text{com } Q_6 = 2.$$

Tendo em conta W , só é possível agregar S_4 e S_6 , resultando o novo conjunto de subrotas:

$$S_1 = \{(11,6); (6,4); (4,3); (3,2); (2,1)\} \quad \text{com } Q_1 = 4$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\} \quad \text{com } Q_2 = 4$$

$$S_3 = \{(11,3); (3,6); (6,5); (5,4); (4,5); (5,11)\} \quad \text{com } Q_3 = 4$$

$$S_4 = \{(11,10); (10,9); (9,8); (8,4); (4,6); (6,11)\} \quad \text{com } Q_4 = 3$$

$$S_5 = \{(11,8); (8,9); (9,10); (10,8); (8,10); (10,11)\} \quad \text{com } Q_5 = 4,$$

que se representa por:

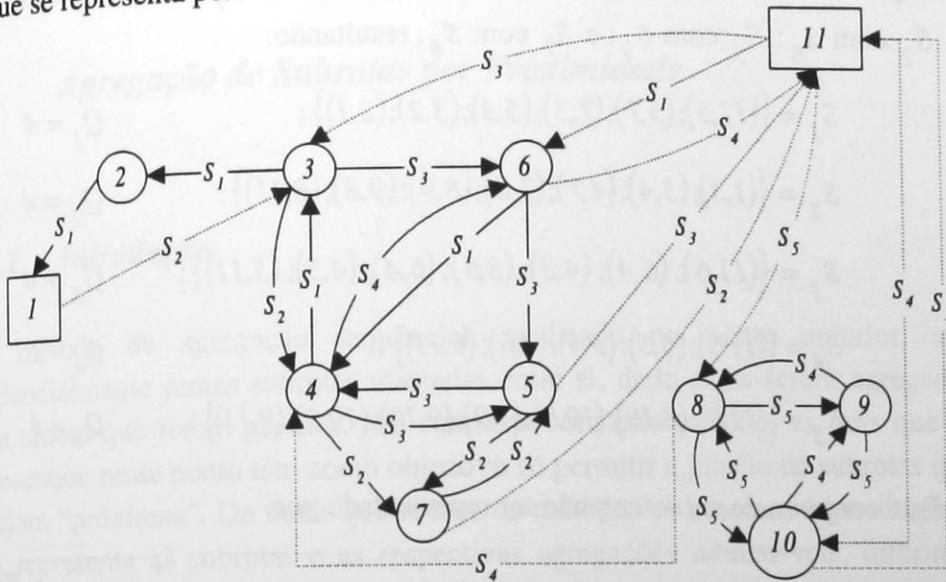


Figura 4.6: Subrotas de GSCE e AgSeq

As únicas duas subrotas agregadas estão afastadas na zona de recolha. Neste exemplo foram geradas duas subrotas com procura total inferior a W por existirem duas componentes conexas, sendo as subrotas mais desequilibradas as últimas geradas em cada componente.

Considerem-se agora as subrotas geradas por GSProc, ou seja:

$$S_1 = \{(11,3); (3,2); (2,1)\} \quad \text{com } Q_1 = 2$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,11)\} \quad \text{com } Q_2 = 2$$

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\} \quad \text{com } Q_3 = 3$$

$$S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\} \quad \text{com } Q_4 = 3$$

$$S_5 = \{(11,5); (5,7); (7,5); (5,11)\} \quad \text{com } Q_5 = 2$$

$$S_6 = \{(11,8); (8,9); (9,8); (8,11)\} \quad \text{com } Q_6 = 2$$

$$S_7 = \{(11,8); (8,10); (10,8); (8,11)\} \quad \text{com } Q_7 = 2$$

$$S_8 = \{(11,9); (9,10); (10,9); (9,11)\} \quad \text{com } Q_8 = 2$$

$$S_9 = \{(11,4); (4,5); (5,11)\} \quad \text{com } Q_9 = 1.$$

Seguindo o método de agregação sequencial, AgSeq, S_1 é agregada com S_5 ; S_2 com S_6 ; S_3 com S_9 ; e S_7 com S_8 , resultando:

$$S_1 = \{(11,5); (5,7); (7,5); (5,3); (3,2); (2,1)\}; \quad Q_1 = 4$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,8); (8,9); (9,8); (8,11)\}; \quad Q_2 = 4$$

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,4); (4,5); (5,11)\}; \quad Q_3 = 4$$

$$S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\}; \quad Q_4 = 3$$

$$S_5 = \{(11,8); (8,10); (10,8); (8,9); (9,10); (10,9); (9,11)\}; \quad Q_5 = 4.$$

Este conjunto de subrotas pode ser representado por:

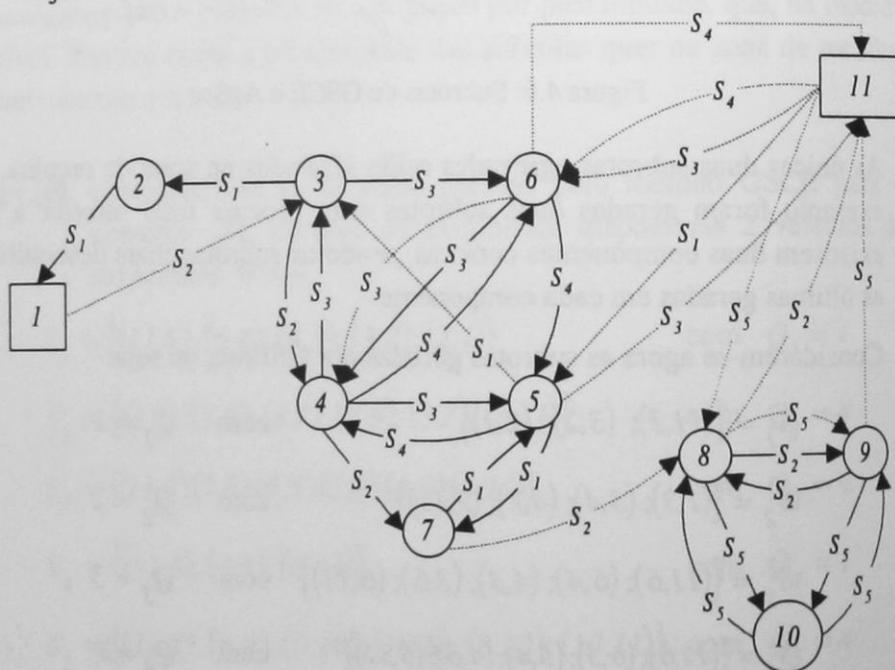


Figura 4.7: Subrotas de GSProc e AgSeq

Como se pode observar as subrotas intersectam-se em diversos pontos, sendo agregadas subrotas afastadas na zona de recolha. Neste exemplo, o conjunto de subrotas resultante de GSCE e AgSeq proporciona a existência de um menor número de caminhos mais curtos em vazio.

Note-se ainda que, por exemplo, na quinta subrota existe um arco em vazio, (8,9), que poderia ser evitado caso a agregação de subrotas fosse possível, não apenas nos extremos, mas a meio. Este tipo de estudos sobre as subrotas construídas será alvo da secção 4.8, dedicada a métodos melhorativos. ♦

4.5. *Agregação de Subrotas por Proximidade*

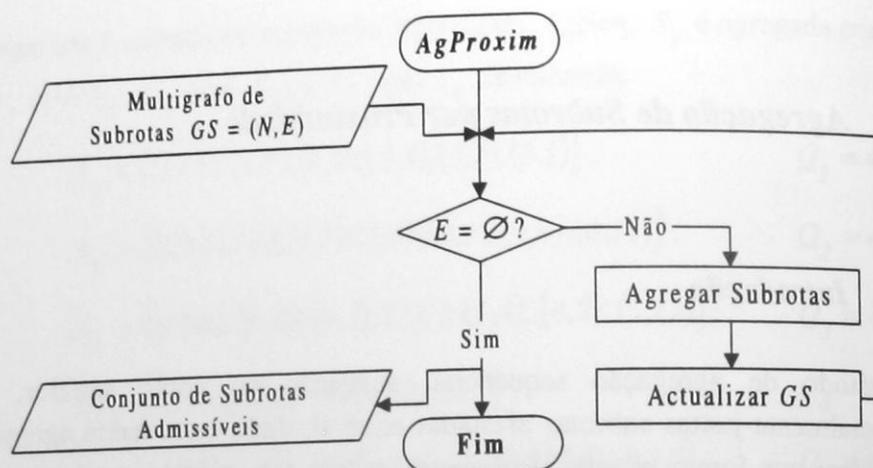
4.5.1. *Introdução*

No método de agregação sequencial, analisado no ponto anterior, são tendencialmente juntas subrotas afastadas entre si, dado estas serem agregadas pela ordem que foram geradas. Em contraste com este método, os dois que se apresentam neste ponto têm como objectivo só permitir a junção de subrotas que estejam "próximas". De notar que ambos os métodos se baseiam no multigrafo que representa as subrotas e as respectivas agregações admissíveis, diferindo essencialmente na forma como são escolhidas as subrotas a agregar.

Num dos métodos, em cada iteração, vão sendo agregados pares de subrotas à custa da determinação de emparelhamentos no multigrafo definido. No segundo método de agregação por proximidade proposto são escolhidas sementes, entre os nodos do multigrafo, que representam subrotas quer afastadas de $n+1$ quer afastadas entre si. Cada uma destas sementes vai originando uma subrota mais próxima de $n+1$ enquanto for possível agregá-la com a subrota que lhe está mais próxima.

Tenta-se assim obter um "bom" critério de agregação. Estando $n+1$ afastado dos vértices do grafo de procura, ou seja, da zona de recolha, a agregação de quaisquer duas subrotas é sempre vantajosa, dado implicar a eliminação de ligações a $n+1$. Escolher a ordem de agregação apenas de acordo com o ordenamento das poupanças resultantes de tal agregação pode ser enganoso, pois comparativamente elas vão ser muito idênticas qualquer que seja o afastamento das subrotas na zona de recolha, dado as distâncias a $n+1$ pesarem muito. Logo, devem ser considerados critérios que se baseiem tanto nas poupanças como no afastamento das subrotas entre si.

Os métodos de agregação por proximidade que se apresentam podem então considerar-se englobados no fluxograma 4.2.



Fluxograma 4.2: Métodos de Agregação por Proximidade – AgProxim

O multigrafo representativo de um conjunto de subrotas define-se no ponto seguinte. Neste multigrafo, cada subrota é representada por um nodo. É criada uma aresta entre dois nodos se for possível agregar as respectivas subrotas, sendo previamente definidos os casos em que se considera possível a junção. Como se verá, os custos associados às arestas estão relacionados com a poupança no valor da solução caso as respectivas subrotas sejam agregadas. Após apresentado o algoritmo utilizado na construção do referido multigrafo, ilustra-se o seu funcionamento com o exemplo que se tem vindo a seguir.

Em 4.5.3. é descrito o primeiro método de agregação por proximidade – *método de emparelhamentos*. Tendo por base o multigrafo de subrotas definido, as subrotas a agregar, em cada iteração, são escolhidas de acordo com as arestas na solução de um emparelhamento. O multigrafo vai sendo actualizado, terminando o processo quando não existirem arestas no multigrafo, ou seja, quando já não for possível agregar mais subrotas.

O terceiro método de agregação, segundo de agregação por proximidade, – *método de sementes* – é apresentado em 4.5.4.. Começando também com o multigrafo de subrotas, este baseia-se numa escolha de sementes. São escolhidos, como se define, nodos do multigrafo (subrotas do problema) para representarem sementes. Definidas as distâncias entre as subrotas e $n+1$, selecciona-se a semente mais afastada de $n+1$, para iniciar o processo iterativo. Enquanto possível, a subrota (semente) seleccionada vai sendo agregada com subrotas que lhe estejam próximas, sendo previamente estabelecido o conceito de proximidade entre subrotas. Sempre que se agregam duas subrotas, o multigrafo é actualizado. O processo repete-se para cada semente considerada e enquanto existirem arestas no multigrafo.

4.5.2. Multigrafo de Subrotas

Definições

Pretende construir-se um multigrafo (GS) que, de algum modo, represente as subrotas e as suas possíveis agregações. Assim, cada subrota é representada por um nodo. Em termos gerais, admite-se a existência de uma aresta, entre dois nodos do multigrafo, se for possível a agregação das respectivas subrotas e se estas forem adjacentes. Usualmente, duas subrotas consideram-se adjacentes se tiverem pelo menos um vértice em comum. Como se verá, nesta tese, duas subrotas serão consideradas adjacentes se tiverem em comum vértices de um certo tipo, e não um qualquer vértice. Esta definição de adjacência proporciona uma implementação computacional mais eficiente.

Caso duas subrotas sejam consideradas adjacentes em dois ou mais pontos, são criadas duas ou mais arestas entre os respectivos nodos, resultando o referido multigrafo. Os custos associados às arestas estão relacionados com as poupanças resultantes da agregação das subrotas em cada um desses pontos, não sendo porém efectuado um estudo muito exaustivo no cálculo destas poupanças. Assim, a poupança resultante da agregação é calculada em função da eliminação de ligações a $n+1$ e de arcos que facilmente sejam identificados como sequência de arcos em vazio na subrota agregada, como se definirá.

Considere-se uma subrota S_i . Represente-se por:

- $V_i = \{i_1, i_2, \dots, i_k\}$ o conjunto de vértices em S_i , diferentes de $n+1$;
- V_i^s o conjunto de vértices em que incide pelo menos um arco percorrido em serviço na subrota S_i , ou seja o conjunto de vértices iniciais ou finais de um desses arcos ($V_i^s \subseteq V_i$);
- V_i^v o conjunto de vértices, diferentes de $n+1$, em que só incidem arcos percorridos em vazio, na subrota S_i , $V_i^v = V_i \setminus V_i^s$.

Note-se que $V_i^s \neq \emptyset$, pois não se admite a existência de subrotas em vazio, ou seja em que não é servido nenhum arco de procura.

Usualmente, neste tipo de problemas, encontra-se a seguinte definição para subrotas adjacentes ([62]):

Definição 4.1: Duas *subrotas*, S_i e S_j , dizem-se *adjacentes* se tiverem pelo menos um vértice, diferente de $n+1$ ⁽¹⁾, em comum, ou seja se $V_i \cap V_j \neq \emptyset$.

Por outro lado, tendo apenas em conta os vértices em que incidem arcos percorridos em serviço, resulta o seguinte conceito de adjacência:

Definição 4.2: Duas *subrotas*, S_i e S_j , dizem-se *adjacentes em serviço* se:

$$V_i^s \cap V_j^s \neq \emptyset.$$

Exemplo 4.1: As subrotas representadas nas figuras seguintes ilustram as diferenças entre as duas definições apresentadas.

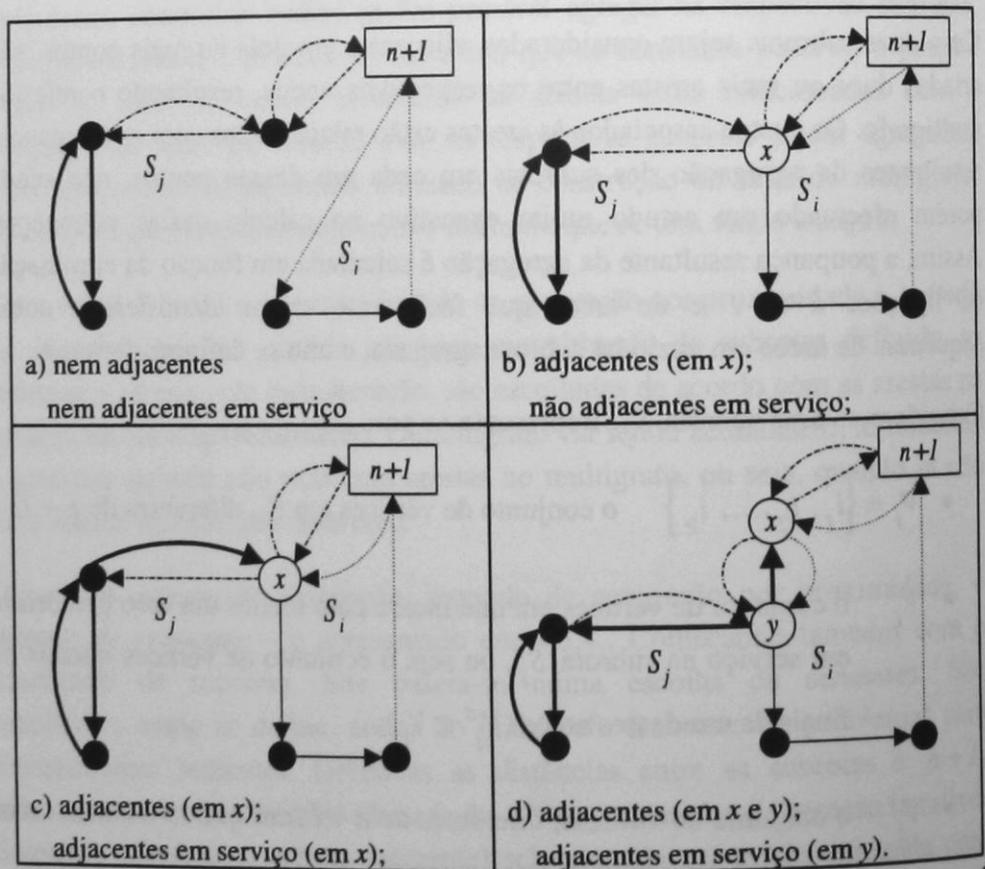


Figura 4.8: Adjacência entre subrotas

⁽¹⁾ O vértice $n+1$ é excluído da definição de adjacência por pertencer a todas as subrotas.

Como é óbvio, $V_i^s \cap V_j^s \subseteq V_i \cap V_j$. Logo, caso $V_i^s \cap V_j^s \subset V_i \cap V_j$, existem mais subrotas adjacentes que adjacentes em serviço.

Note-se que os conjuntos de subrotas adjacentes que se obtêm com estas duas definições coincidem sse $V_i^s \cap V_j^s = V_i \cap V_j$.

Representando as arestas do multigrafo GS subrotas adjacentes que se podem agregar, também este conjunto de arestas difere em função da definição utilizada, sabendo-se que, para um mesmo conjunto de subrotas, um contém o outro.

Estando o conjunto de arestas do multigrafo inicial, se utilizada a definição de adjacência em serviço (definição 4.2), contido no conjunto de arestas do multigrafo inicial se utilizada a definição de adjacência (definição 4.1), pode provar-se dominância da definição 4.1 em relação à definição 4.2, na primeira iteração dos algoritmos de agregação por proximidade. Ou seja, se com base em cada uma das definições se definir o multigrafo de subrotas, para um mesmo conjunto de subrotas, obtêm-se dois multigrafos $G1$ e $G2$. Agregando subrotas em cada um destes multigrafos, com base na primeira iteração de um mesmo método, a solução correspondente a agregar em $G1$ tem um valor nunca superior à correspondente a agregar as subrotas em $G2$. Tal deve-se ao facto de, inicialmente, as poupanças nunca serem piores quando utilizada a definição de adjacência. Porém, tal dominância não se reflecte obrigatoriamente na solução final. No fim da primeira iteração, sendo em geral diferentes as subrotas agregadas dependendo da definição utilizada, vem diferente o novo multigrafo resultante do passo de actualização.

Nesta tese foi adoptada a definição de adjacência em serviço. Assim, doravante, sempre que se fale de subrotas adjacentes está a referir-se a subrotas adjacentes em serviço. Tal é motivado pelo facto desta definição permitir implementações mais eficientes, pensando-se mesmo ser vantajoso, em geral e com vista à agregação, não detectar a adjacência entre subrotas que o sejam só em vazio, não sendo assim possível a agregação de tais subrotas. A agregação de subrotas deste tipo originaria subrotas com um maior número de interrupções de serviço face a agregações de subrotas adjacentes apenas em serviço.

Tendo em conta esta definição de adjacência, na representação das subrotas, caminhos em vazio são substituídos por arcos de procura nula entre os extremos do caminho. Mantém-se assim a notação que tem sido considerada, em que o arco (x, y) , percorrido em vazio, representa o caminho de menor custo de x para y , em G , com custo igual a $spl(x, y)$. Tal representação, dado não exigir a identificação dos arcos nos caminhos em vazio, conduz a uma implementação

mais eficiente, quer em termos de memória ocupada quer em termos de tempo de execução.

Nestes métodos a agregação de subrotas, sendo esta permitida apenas entre subrotas adjacentes em serviço, é obviamente feita em vértices comuns às duas subrotas. Considerem-se então as subrotas intermédias:

$$S_i = \left\{ (n+1, i_1, \dots, i_\ell, x, i_{\ell+1}, \dots, i_r, n+1) \right\}$$

e

$$S_j = \left\{ (n+1, j_1, \dots, j_k, x, j_{k+1}, \dots, j_m, n+1) \right\}.$$

Incluir S_j em S_i , resulta, em termos genéricos, ou seja, sem eliminação de eventuais seqüências de arcos em vazio, a subrota:

$$(i) \left\{ (n+1, i_1, \dots, i_\ell, x, j_{k+1}, \dots, j_m, j_1, \dots, j_k, x, i_{\ell+1}, \dots, i_r, n+1) \right\}$$

Incluir S_i em S_j , origina:

$$(ii) \left\{ (n+1, j_1, \dots, j_k, x, i_{\ell+1}, \dots, i_r, i_1, \dots, i_\ell, x, j_{k+1}, \dots, j_m, n+1) \right\}.$$

Doravante considera-se que $\overline{\overline{(x, y)}}$ representa que o arco (x, y) está a ser atravessado em vazio, sempre que seja necessário referir este facto.

Definição 4.3: A poupança de agregar as subrotas intermédias S_i e S_j no vértice comum x , é:

$$s_{i,j}(x) = \text{Max} \left\{ 0; s_{i \oplus j}(x); s_{j \oplus i}(x) \right\}$$

onde,

$$s_{i \oplus j}(x) = \text{spl}(n+1, j_1) + \text{spl}(j_m, n+1) - \text{spl}(j_m, j_1) + \beta;$$

$$\beta = \begin{cases} \text{spl}(i_\ell, x) + \text{spl}(x, j_{k+1}) - \text{spl}(i_\ell, j_{k+1}) & \text{se } \overline{\overline{(i_\ell, x)}} \text{ e } \overline{\overline{(x, j_{k+1})}} \\ \text{spl}(j_k, x) + \text{spl}(x, i_{\ell+1}) - \text{spl}(j_k, i_{\ell+1}) & \text{se } \overline{\overline{(j_k, x)}} \text{ e } \overline{\overline{(x, i_{\ell+1})}} \\ 0 & \text{c.c.} \end{cases}$$

$$s_{j \oplus i}(x) = \text{spl}(n+1, i_1) + \text{spl}(i_r, n+1) - \text{spl}(i_r, i_1) + \beta.$$

Se S_i for uma subrota inicial (final) só se admite a inclusão de S_j em S_i . A poupança de agregar a subrota inicial (final), S_i , com a subrota intermédia, S_j , no vértice comum x , é definida por $Max \{0; s_{i \oplus j}(x)\}$.

Por vezes o valor calculado como poupança é inferior à poupança efectiva da junção das subrotas. Tal como se, por exemplo, em (i) (j_m, j_l) for antecedido e/ou sucedido por arcos percorridos em vazio. O vértice, x , comum a duas subrotas pode também estar repetido numa das subrotas, ou mesmo nas duas. Neste caso o cálculo das poupanças é feito para a primeira vez que o vértice é considerado na subrota. Não é portanto efectuado um estudo exaustivo no cálculo das poupanças.

Algoritmo para Construção do Multigrafo de Subrotas

O algoritmo GS, que se apresenta de seguida, sintetiza a forma de construção do multigrafo de subrotas, tendo em conta a definição de adjacência adoptada. Como referido, os dois métodos de agregação por proximidade baseiam-se neste multigrafo.

Pelas razões apresentadas no método de agregação sequencial, AgSeq, só é possível agregar duas subrotas se pelo menos uma for intermédia e se a capacidade dos veículos o permitir. Assim, só nestes casos e se as subrotas forem adjacentes se considera uma aresta entre os nodos do multigrafo.

Tal como anteriormente, a agregação de uma subrota inicial (final) com uma subrota intermédia originará sempre uma subrota inicial (final). De facto dado serem necessárias exactamente P subrotas iniciais e P subrotas finais para construção da solução admissível, não faz sentido considerar estas agregações de outra forma.

Algoritmo GS: Construção do Multigrafo de Subrotas $GS = (N, E)$

0. Seja:
- n_subr o número de subrotas;
 - S_k a sequência de arcos na subrota k ;
 - Q_k a procura total em S_k ;

Inicializar: $E = \emptyset$; {conjunto de arestas de GS}

1. Nodos: Considerar que cada subrota é representada por um nodo do multigrafo, i.e., $N = \{n_1, n_2, \dots, n_{n_subr}\}$;
2. Arestas: Se S_i e S_j são adjacentes em ℓ vértices distintos e podem ser agregadas, considerar ℓ arestas entre os dois nodos respectivos, n_i e n_j , uma por cada um dos vértices em comum nas duas subrotas. Ou seja,

$$\text{Se } |V_i^s \cap V_j^s| = \ell > 0$$

$$\text{e } Q_i + Q_j \leq W$$

e pelo menos uma das subrotas é intermédia

então considerar, em E , ℓ réplicas de (n_i, n_j) .

Repetir 2. para cada par de nodos de N ;

3. Custos: Com base na definição 4.3 calcular o custo de cada réplica da aresta (n_i, n_j) , $s_{i,j}(x)$, associado à poupança no valor da solução resultante da agregação das subrotas, S_i e S_j , no vértice comum x ;

Se $s_{i,j}(x) = 0$ retirar a aresta do grafo, pois a agregação das respectivas subrotas originaria uma solução de valor nunca inferior ao da actual;

Repetir 3. para cada aresta de E ;

4. Retirar de N todos os nodos de grau nulo em GS .

Fim.

Constrói-se assim um multigrafo, em que cada nodo representa uma subrota. Uma aresta entre dois nodos representa uma possível agregação entre duas subrotas. A ideia de construção de grafos representativos de ciclos, em problemas similares ao PRRS, foi inicialmente proposta por Male e Liebman [62], embora os grafos de ciclos construídos por estes autores não se assemelhem com os que se obtêm pelo algoritmo GS.

Exemplo A: Retomem-se as subrotas geradas por GSCE para este exemplo. Como se viu no método de agregação sequencial, as únicas subrotas passíveis de serem agregadas, tendo em conta W , eram S_4 e S_6 . Porém estas duas subrotas não tendo vértices em comum, não são consideradas adjacentes. Logo, GS não tem arestas, não sendo portanto, neste caso, aplicáveis os métodos que se baseiam em GS .

Considerem-se agora as subrotas geradas por GSProc:

$$S_1 = \{(11,3); (3,2); (2,1)\} \quad \text{com } Q_1 = 2$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,11)\} \quad \text{com } Q_2 = 2$$

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\} \quad \text{com } Q_3 = 3$$

$$S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\} \quad \text{com } Q_4 = 3$$

$$S_5 = \{(11,5); (5,7); (7,5); (5,11)\} \quad \text{com } Q_5 = 2$$

$$S_6 = \{(11,8); (8,9); (9,8); (8,11)\} \quad \text{com } Q_6 = 2$$

$$S_7 = \{(11,8); (8,10); (10,8); (8,11)\} \quad \text{com } Q_7 = 2$$

$$S_8 = \{(11,9); (9,10); (10,9); (9,11)\} \quad \text{com } Q_8 = 2$$

$$S_9 = \{(11,4); (4,5); (5,11)\} \quad \text{com } Q_9 = 1.$$

Neste caso, no final do passo 2., GS vem igual a:

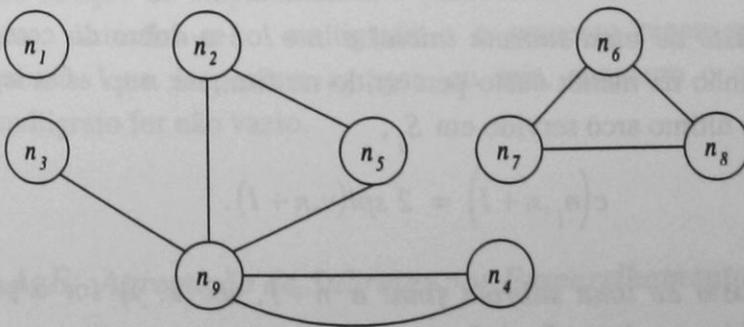


Figura 4.9: Grafo GS para as subrotas de GSProc

Vão então definir-se, sobre GS , processos de agregação das subrotas duas a duas. Assim, tendo em conta os custos das arestas de GS , associados às poupanças de agregar as subrotas adjacentes, há que definir a agregação de subrotas, ou seja, há que seleccionar arestas de GS .

4.5.3. Método de Emparelhamentos

Descrição do Método

Neste método as subrotas a agregar são escolhidas de acordo com soluções heurísticas de emparelhamentos de custo máximo no multigrafo GS , definido no ponto anterior. Construído o multigrafo que representa as subrotas e as respectivas agregações possíveis, vão agregar-se em cada iteração, pares de subrotas. Estando os custos associados às arestas relacionados com as poupanças resultantes da agregação, há que tentar maximizar a poupança.

Um dos critérios usados em caso de empate, na escolha das subrotas a agregar, está relacionado com o custo das subrotas a $n+1$. Este conceito é exposto de seguida.

Definição 4.4:

- O *custo de uma subrota intermédia a $n+1$* , é a soma dos custos dos caminhos de menor custo de ligação desta subrota a $n+1$. Assim, se (x, y) e (u, v) forem, respectivamente, o primeiro e o último arco servido na subrota S_i , define-se:

$$c(n_i, n+1) = spl(n+1, x) + spl(v, n+1),$$

onde n_i é o vértice que representa S_i em GS .

- O *custo de uma subrota inicial a $n+1$* é o dobro do comprimento do caminho de menor custo percorrido no fim, até $n+1$. Ou seja, se (u, v) for o último arco servido em S_i ,

$$c(n_i, n+1) = 2 spl(v, n+1).$$

- O *custo de uma subrota final a $n+1$* , se (x, y) for o primeiro arco servido na subrota final S_i , é:

$$c(n_i, n+1) = 2 spl(n+1, x).$$

Numa iteração, a maximização da poupança associada à agregação de pares de subrotas consegue alcançar-se se determinados os emparelhamentos de peso máximo. Podendo não existir um número par de nodos em GS , não se pode falar de emparelhamentos perfeitos.

Como é sabido os métodos existentes para determinar emparelhamentos são pesados e morosos, embora polinomiais.

Por outro lado, a maximização das poupanças, em cada iteração, não garante a maximização da poupança total com a agregação. Verifica-se, em geral, que o esforço necessário para a resolução exacta dos emparelhamentos máximos não compensa, face à qualidade das soluções que se obtêm quando determinados emparelhamentos de uma forma heurística. Uma heurística simples produz soluções com um esforço computacional muito inferior, não sendo necessariamente piores que as resultantes dos emparelhamentos máximos. Tal a razão de recorrer a soluções heurísticas, mesmo tratando-se de um problema polinomial.

Determina-se então, de forma heurística, um emparelhamento no multigrafo de subrotas definido. Nesta heurística, começa-se por ordenar as arestas por ordem decrescente das poupanças. Em caso de empate, considera-se, caso exista a aresta:

- 1º) que tenha mais réplicas, pois está associada a subrotas adjacentes em mais de um ponto;
- 2º) incidente num vértice de menor grau, pois corresponde a uma subrota que tem menos alternativas para ser agregada;
- 3º) que estiver associada à agregação de subrotas mais afastadas de $n + 1$.

Respeitando a definição de emparelhamento, as arestas vão sendo marcadas de acordo com a sua sequência na lista ordenada.

As arestas na solução do emparelhamento (marcadas) indicam as subrotas que são agregadas. Actualiza-se o multigrafo e o processo repete-se enquanto existirem subrotas que se possam agregar, ou seja, enquanto o conjunto de arestas do multigrafo for não vazio.

Algoritmo AgE: Agregação de Subrotas por Emparelhamentos

0. Seja: n_subr o número de subrotas;
 S_k a sequência dos arcos incluídos na subrota k ;
1. Construir o multigrafo GS (algoritmo GS);
 Considerar todos os vértices de GS e todas as arestas por marcar;
2. Se $E = \emptyset$ **Fim**;

3. Encontrar um emparelhamento em GS

3.1 Ordenar, numa lista $E = [e_1, e_2, \dots]$, as arestas ⁽¹⁾ por ordem decrescente das poupanças, $s_{i,j}(x)$. Em caso de empate considerar primeiro as arestas com um maior número de réplicas, incidentes em vértices de menor grau e/ou mais afastados de $n+1$;

3.2. Fazer: $r \leftarrow 1$; {índice, em E , da aresta em análise}
 $n_agregar \leftarrow 0$; {n.º de pares de subrotas a agregar}

3.3. Enquanto $(i_r$ ou j_r marcados) e $(r \leq n_subr)$
 Fazer $r \leftarrow r+1$;

3.4. Se $r \leq n_subr$
 Marcar: e_r, i_r e j_r
 Fazer $n_agregar \leftarrow n_agregar + 1$;

3.5. Se $(n_agregar = n_subr \text{ div } 2)$ ou $(r > n_subr)$
 Ir para 4.;

C.c., Fazer $r \leftarrow r+1$;

Voltar a 3.3.;

4. Agregar as subrotas correspondentes às arestas marcadas;

Actualizar o conjunto de subrotas e n_subr ;

Voltar a 1..

Obviamente que este método proporcionando, em geral, um melhor conjunto de subrotas que AgSeq sobre o conjunto de subrotas geradas por GSProc, não terá vantagens significativas se utilizado sobre o conjunto de subrotas geradas por GSCE. Conforme frisado, este conjunto não deve proporcionar muitas agregações, sendo que as subrotas passíveis de vir a ser agregadas não devem representar, em geral, subrotas adjacentes, pois, na sua maioria, resultarão de componentes conexas distintas. Assim, AgE é apenas considerado para agregar as subrotas resultantes de GSProc, dando origem à heurística H2.

(1) Onde cada aresta representa a agregação de duas subrotas (nodos da aresta). Assume-se que $e_r = (i_r, j_r)$, representando a possível agregação das subrotas S_{i_r} e S_{j_r} .

Exemplo A: Considere-se o grafo GS resultante do conjunto de subrotas geradas por $GSProc$ para o exemplo protótipo, onde a numeração sobre as arestas indica a ordenação destas por ordem decrescente de poupanças, pretendida em 3.1.

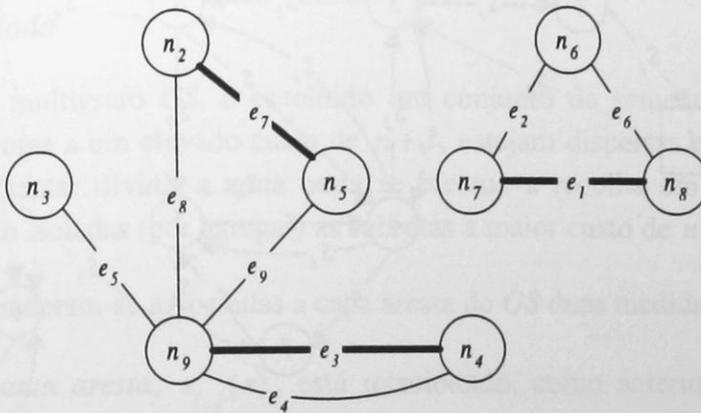


Figura 4.10: Grafo GS

De acordo com AgE o emparelhamento neste grafo é formado pelas arestas:

$$E = \{e_1; e_3; e_7\}.$$

Logo, vão ser agregadas as subrotas: S_7 e S_8 ; S_9 e S_4 ; S_2 e S_5 , resultando o conjunto de subrotas:

$$S_1 = \{(11,3); (3,2); (2,1)\}, \quad Q_1 = 2;$$

$$S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\}, \quad Q_2 = 4;$$

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\}, \quad Q_3 = 3;$$

$$S_4 = \{(11,6); (6,5); (5,4); (4,5); (5,4); (4,6); (6,11)\}, \quad Q_4 = 4;$$

$$S_5 = \{(11,8); (8,9); (9,8); (8,11)\}, \quad Q_5 = 2;$$

$$S_6 = \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,11)\}, \quad Q_6 = 4.$$

que se pode representar por:

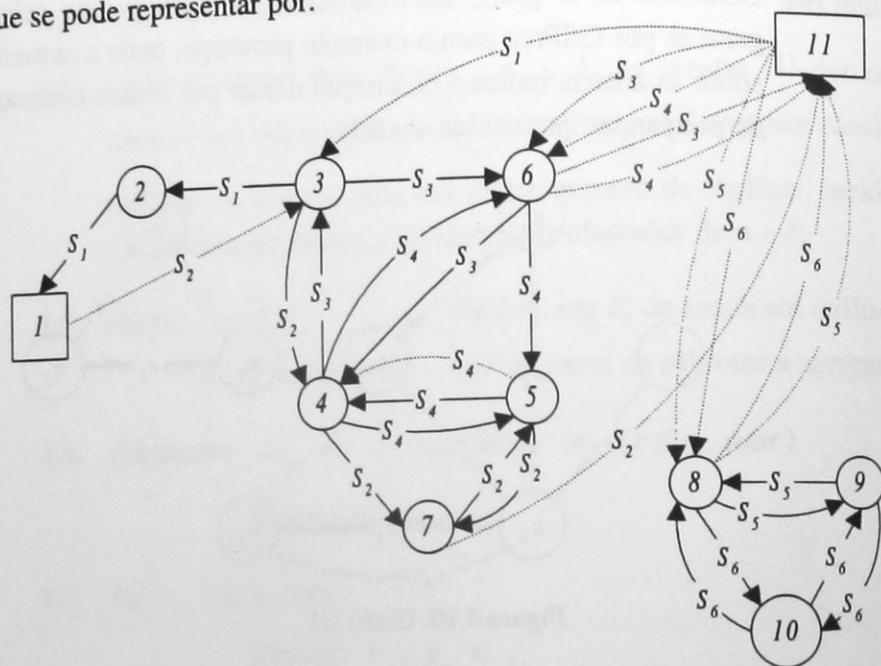


Figura 4.11: Subrotas admissíveis de GSProc e AgE

Este é o conjunto final de subrotas obtido com AgE, pois entre as subrotas que poderiam ser agregadas, por ser compatível com W , não existem subrotas adjacentes. Logo, GS não tem arestas e o algoritmo termina. ♦

Como se viu, a agregação de duas subrotas considera-se possível se compatível com a capacidade dos veículos. Note-se que, sendo agregados pares de subrotas em cada iteração, pode obter-se um conjunto de subrotas em que cada uma ocupa cerca de 51% da capacidade dos veículos. Soluções com estas características, dificilmente terão um valor próximo do valor óptimo. Como é usual neste tipo de abordagens, tal poderia tentar evitar-se se, por exemplo, nas primeiras iterações, só fosse permitida a agregação de duas subrotas caso a soma das respectivas procuras não excedesse $W/2 - \theta$, com θ fixo.

4.5.4. Método de Sementes

Descrição do Método

Tendo por base o multigrafo GS , é escolhido um conjunto de sementes que representando subrotas a um elevado custo de $n+1$, estejam dispersas em GS . Pretende-se assim tentar dividir a zona onde se efectua a recolha em várias zonas, não deixando isoladas (por agregar) as subrotas a maior custo de $n+1$.

Antes de mais, consideram-se associadas a cada aresta de GS duas medidas:

- O **custo de uma aresta**, $s_{i,j}(x)$, está relacionado, como anteriormente (definição 4.1), com a poupança de agregar as subrotas S_i e S_j , representadas pelos nodos da aresta, n_i e n_j , no vértice comum x .
- Uma **distância unitária** associada a cada aresta de GS é utilizada para determinar o afastamento entre subrotas, que se define de seguida.

Definição 4.5: *Distância entre subrotas* é o comprimento do caminho mais curto entre os nodos de GS que as representam, assumindo arestas de distâncias unitárias. Representa-se por $d(n_i, n_j)$ a distância entre as subrotas S_i e S_j .

A distância entre duas subrotas representa então, o número mínimo de arestas que separam os respectivos nodos de GS , como se exemplifica de seguida.

Exemplo 4.2: Considere-se o grafo GS :

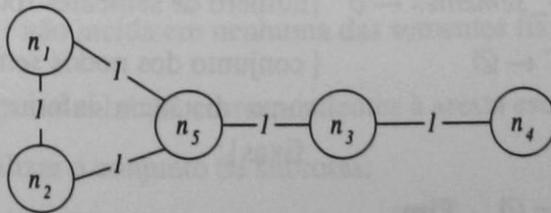


Figura 4.12: Distâncias definidas no grafo GS

Onde, por exemplo, $d(n_1, n_2) = 1$; $d(n_1, n_3) = 2$; $d(n_1, n_4) = 3$. ♦

Neste método de agregação por proximidade começa por escolher-se um conjunto de sementes, tendo por base as definições 4.4 e 4.5. Assim, conhecendo $K + P$, o número mínimo de subrotas, escolhem-se $K + P - \alpha$ ⁽¹⁾ sementes, nodos do multigrafo, representando subrotas que estando a maior custo de $n + 1$, estejam afastadas entre si. Existindo em GS pelo menos d ⁽²⁾ arestas a separar duas subrotas sementes, talvez seja possível agregar todas as sementes com subrotas adjacentes. Escolhendo para sementes as subrotas a maior custo de $n + 1$, tenta-se não deixar por agregar as respectivas subrotas, pois estas são as mais penalizadoras em termos do custo de ligações a $n + 1$ e, conseqüentemente, do valor da solução.

Como será referido, sempre que na escolha das sementes existir empate, opta-se por seleccionar a de menor grau. Mais uma vez, tenta-se não isolar subrotas entre as mais afastadas de $n + 1$.

Escolhidas as sementes, inicia-se o processo de escolha de subrotas a agregar (arestas de GS). Se possível, começa-se por agregar à subrota a maior custo de $n + 1$, primeira semente escolhida, subrotas a uma distância nunca inferior a d (fixo) das restantes sementes. Quando já não é possível agregar nenhuma outra subrota a esta, considera-se a segunda semente. O processo repete-se enquanto existirem subrotas que possam ser agregadas, como se apresenta no algoritmo AgS.

Algoritmo AgS: Agregação de Subrotas com Definição de Sementes

0. Dados: α constante inteira para determinar o número de sementes;
 d distância mínima entre sementes, assumindo todas as arestas de GS com distância unitária;
1. Construir o multigrafo $GS = (N, E)$ (algoritmo GS);
 Fazer: $n_sementes \leftarrow 0$ {número de sementes fixas};
 $\Gamma \leftarrow \emptyset$ {conjunto dos nodos sementes e dos nodos a uma distância inferior a d das sementes fixas};
2. Se $E = \emptyset$ Fim;

⁽¹⁾ sendo α um número inteiro e "pequeno", em comparação com o valor de $K + P$

⁽²⁾ $d \geq 1$ é escolhido em função da distância calculada entre subrotas, nodos de GS .

3. Fixar $K + P - \alpha$ sementes, no máximo:

3.1. Entre os nodos de GS ainda não analisadas e não pertencentes a Γ , seleccionar o mais afastado de $n+1$. Em caso de empate seleccionar o de menor grau;

Fixar como semente o nodo seleccionado;

Fazer: $n_sementes \leftarrow n_sementes + 1$;

Actualizar Γ , incluindo a semente fixa e todos os nodos de GS a uma distância inferior a d desta semente;

3.2. Se $(n_sementes < K + P - \alpha)$ e (existirem nodos por analisar)

Voltar a 3.1.;

4. Ordenar as sementes por ordem decrescente do seu afastamento a $n+1$;

Seja: $n_{k_1}, n_{k_2}, \dots, n_{k_{n_sementes}}$, a lista ordenada de sementes;

Fazer: $s \leftarrow 1$;

5. Se $s = n_sementes$ Fazer $\Gamma \leftarrow \emptyset$

C.c., Retirar de Γ a semente n_{k_s}

Retirar de Γ todos os nodos que só relativamente à semente n_{k_s} é que estão a uma distância inferior a d ;

6. Agregar S_{k_s} com subrotas adjacentes:

Se existirem arestas incidentes em n_{k_s} e não incidentes em nodos de Γ , escolher a aresta de maior custo (poupança) associado;

Se todas as arestas incidentes em n_{k_s} incidem também em vértices de Γ , escolher, entre estas, a de maior custo que não incida em nenhuma das sementes fixas;

Agregar as subrotas correspondentes à aresta escolhida;

Actualizar o conjunto de subrotas;

Actualizar GS , analisando se é preciso retirar arestas e/ou nodos do multigrafo (por exemplo, por causa da procura/capacidade);

7. *Se* ainda existem arestas incidentes em n_{k_s} , **Voltar a 6.**

C.c., Fazer: $s \leftarrow s + 1$;

Se $s \leq n_sementes$ **Voltar a 5.;**

C.c., **Ir para 8.;**

8. *Se* $E \neq \emptyset$

Se $|E| = 1$ Agregar as subrotas correspondentes a esta aresta

Fim

C.c., Escolher uma nova semente:

Entre os nodos de grau não nulo, escolher, como semente, o de maior custo a $n + 1$. Se existir mais de um vértice nestas condições, seleccionar o de menor grau;

Fazer: $n_sementes \leftarrow n_sementes + 1$;

Voltar a 6.;

C.c., **Fim.**

No passo 6. é escolhida a aresta de maior poupança associada para identificar as duas subrotas a agregar. Formas alternativas para a escolha das arestas proporcionam, obviamente, diferentes soluções. Nesta escolha poder-se-ia ter considerado uma relação entre as poupanças e as capacidades dos veículos. Assim, poderia ter-se escolhido, por exemplo, a aresta de maior poupança associada se a procura nas subrotas não é superior a metade da capacidade dos veículos e a de menor poupança associada (representando, em princípio, uma subrota mais próxima de $n + 1$ entre as adjacentes à semente), nos restantes casos.

Pelas razões apresentadas no método de agregação anterior, também não faz sentido aplicar este método sobre o conjunto de subrotas proveniente de GSCE. Assim, este método só é utilizado sobre o conjunto de subrotas gerado por GSProc, originando a heurística H3.

Exemplo A: Retome-se o grafo GS resultante do conjunto de subrotas geradas por GSProc para este exemplo, onde a numeração das arestas corresponde à ordenação por ordem decrescente dos custos associados (poupanças).

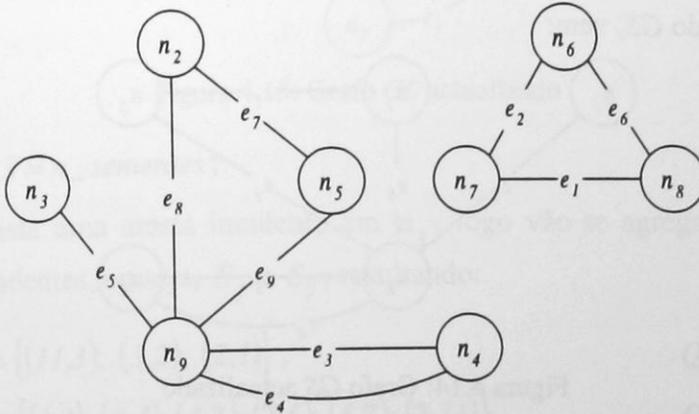


Figura 4.13: Grafo GS para o exemplo **A**

3. Fixação de $K + P - \alpha = 3 + 2 - 2 = 3$ sementes, sendo dado $d = 1$.

Considere-se que os vértices no seguinte conjunto estão ordenados por ordem decrescente do seu custo a $n + 1$: $[n_7; n_8; n_2; n_5; n_9; n_6; n_3; n_4]$.

Neste caso a primeira semente é n_7 e $\Gamma = \{n_7; n_6; n_8\}$.

A segunda semente é n_2 , resultando $\Gamma = \{n_7; n_6; n_8; n_2; n_5; n_9\}$.

A terceira semente é n_3 , resultando $\Gamma = \{n_7; n_6; n_8; n_2; n_5; n_9; n_3\}$.

Foram assim fixas 3 sementes.

5. De acordo com a lista ordenada de nodos, n_7 é a primeira semente, logo

$$\Gamma = \{n_2; n_5; n_9; n_3\}.$$

6. Entre as arestas incidentes em n_7 (n_7, n_8) é a de maior poupança, logo vão-se agregar as subrotas S_7 e S_8 , resultando:

$S_1 = \{(11,3); (3,2); (2,1)\},$	$Q_1 = 2;$
$S_2 = \{(1,3); (3,4); (4,7); (7,11)\},$	$Q_2 = 2;$
$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\},$	$Q_3 = 3;$
$S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\},$	$Q_4 = 3;$
$S_5 = \{(11,5); (5,7); (7,5); (5,11)\},$	$Q_5 = 2;$

$$\begin{aligned}
 S_6 &= \{(11,8); (8,9); (9,8); (8,11)\}, & Q_6 &= 2; \\
 S_7 &= \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,11)\}, & Q_7 &= 4; \\
 S_8 &= \{(11,4); (4,5); (5,11)\}, & Q_8 &= 1.
 \end{aligned}$$

Actualizando GS, vem:

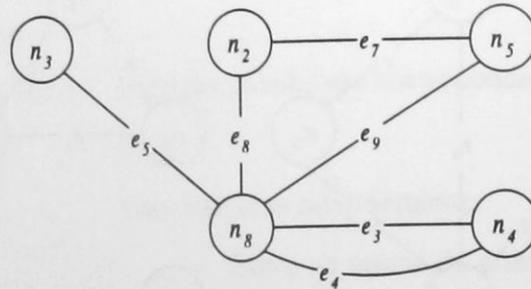


Figura 4.14: Grafo GS actualizado

7. $s \leftarrow 2 < n_{\text{sementes}}$;

5. De acordo com a lista ordenada de nodos, n_2 é a segunda semente, logo:

$$\Gamma = \{n_3; n_8\};$$

6. Entre as arestas incidentes em n_2 , (n_2, n_5) é a de maior poupança, logo vão-se agregar as subrotas S_2 e S_5 , resultando:

$$\begin{aligned}
 S_1 &= \{(11,3); (3,2); (2,1)\}, & Q_1 &= 2; \\
 S_2 &= \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\}, & Q_2 &= 4; \\
 S_3 &= \{(11,6); (6,4); (4,3); (3,6); (6,11)\}, & Q_3 &= 3; \\
 S_4 &= \{(11,6); (6,5); (5,4); (4,6); (6,11)\}, & Q_4 &= 3; \\
 S_5 &= \{(11,8); (8,9); (9,8); (8,11)\}, & Q_5 &= 2; \\
 S_6 &= \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,11)\}, & Q_6 &= 4; \\
 S_7 &= \{(11,4); (4,5); (5,11)\}, & Q_7 &= 1.
 \end{aligned}$$

Actualizando GS, vem:

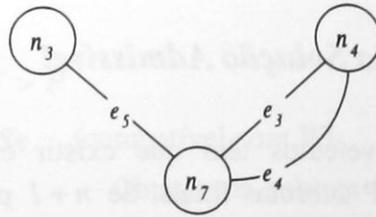


Figura 4.15: Grafo GS atualizado

7. $s \leftarrow 3 = n_sementes$;

6. Só existe uma aresta incidente em n_3 , logo vão-se agregar as subrotas correspondentes, ou seja, S_3 e S_7 , resultando:

- | | |
|---|-------------|
| $S_1 = \{(11,3); (3,2); (2,1)\}$, | $Q_1 = 2$; |
| $S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\}$, | $Q_2 = 4$; |
| $S_3 = \{(11,6); (6,4); (4,5); (5,4); (4,3); (3,6); (6,11)\}$, | $Q_3 = 4$; |
| $S_4 = \{(11,6); (6,5); (5,4); (4,6); (6,11)\}$, | $Q_4 = 3$; |
| $S_5 = \{(11,8); (8,9); (9,8); (8,11)\}$, | $Q_5 = 2$; |
| $S_6 = \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,11)\}$, | $Q_6 = 4$. |

que se pode representar por:

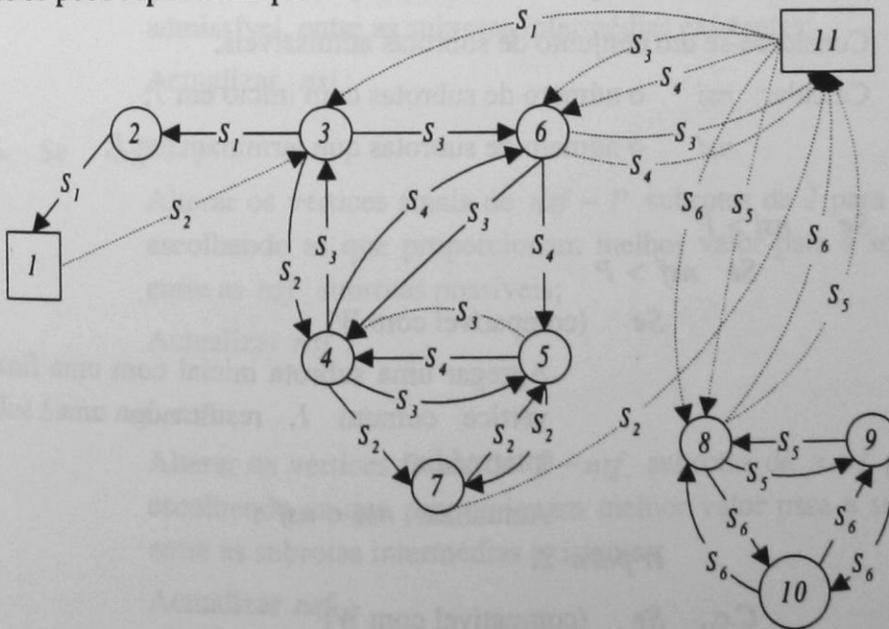


Figura 4.16: Subrotas admissíveis de GSProc e AgS

Note-se que relativamente ao conjunto de subrotas obtido por GSProc e AgE apenas as subrotas S_3 e S_4 são distintas. ◆

4.6. *Geração de uma Solução Admissível*

Existindo disponíveis P veículos têm que existir exactamente P subrotas iniciais, de 1 para $n+1$, P subrotas finais, de $n+1$ para 1 , representando as restantes subrotas intermédias.

Assim, para finalmente gerar uma solução admissível, basta, eventualmente, alterar o início e/ou o fim de algumas subrotas para que passem a existir tantas subrotas iniciais e finais quantos os veículos.

O algoritmo SolAdm, que se descreve de seguida, permite construir uma solução admissível a partir de um conjunto de subrotas admissíveis. Nesta fase, considerou-se também a eventual agregação entre subrotas iniciais e subrotas finais, caso seja compatível com a capacidade dos veículos e vantajoso em função do número de subrotas iniciais e finais existentes face ao desejado.

Na implementação computacional admitiu-se ainda, no final, a agregação entre subrotas que não sendo adjacentes possam e devam ser agregadas.

Algoritmo SolAdm: Geração de uma Solução Admissível

0. Considere-se um conjunto de subrotas admissíveis;

Calcular: nsi o número de subrotas com início em 1 ;

nsf o número de subrotas que terminam em 1 ;

1. Se $nsi > P$

Se $nsf > P$

Se (compatível com W)

Agregar uma subrota inicial com uma final no vértice comum 1 , resultando uma subrota intermédia;

Actualizar nsi e nsf ;

Ir para 3.

C.c., Se (compatível com W)

Obter uma subrota inicial, agregando duas subrotas iniciais, considerando uma seguida da outra e escolhendo a melhor das duas alternativas possíveis;

Actualizar nsi ;

Ir para 3.2. Se $nsi < P$ Se $nsf > P$ Se (compatível com W)

Obter uma subrota final da agregação de duas subrotas finais, considerando uma seguida da outra e escolhendo a melhor das duas alternativas possíveis;

Actualizar nsf ;

3. **Repetir 1. e 2.** enquanto existirem subrotas iniciais ou finais por analisar, $nsi \neq P$ e $nsf \neq P$;

4. Se $nsi > P$

Alterar os vértices iniciais de $nsi - P$ subrotas de l para $n+1$, escolhendo as que proporcionam melhor valor para a solução, entre as nsi subrotas possíveis;

Actualizar nsi ;

5. Se $nsi < P$

Alterar os vértices iniciais de $P - nsi$ subrotas de $n+1$ para l , escolhendo as que proporcionam melhor valor para a solução admissível, entre as subrotas intermédias existentes;

Actualizar nsi ;

6. Se $nsf > P$

Alterar os vértices finais de $nsf - P$ subrotas de l para $n+1$, escolhendo as que proporcionam melhor valor para a solução, entre as nsf subrotas possíveis;

Actualizar nsf ;

7. Se $nsf < P$

Alterar os vértices finais de $P - nsf$ subrotas de $n+1$ para l , escolhendo as que proporcionam melhor valor para a solução, entre as subrotas intermédias existentes;

Actualizar nsf ;

8. Calcular o valor da solução admissível, dado por:

$$\bar{z} = C_{A_R} + \sum spl(i, j),$$

onde (i, j) representam os arcos percorridos em vazio na solução admissível gerada.

Os métodos de geração e agregação são então sucedidos por SolAdm, para serem obtidas soluções admissíveis para o PRRS, na terceira fase das heurísticas. Tendo em conta as possíveis conjugações consideradas entre os diferentes métodos, ficam então disponíveis as três heurísticas:

- H1 correspondente à sequência de métodos GSCE, AgSeq e SolAdm;
- H2 correspondente à sequência de métodos GSProc, AgE e SolAdm;
- H3 correspondente à sequência de métodos GSProc, AgS e SolAdm.

Exemplo A: Considere-se o conjunto de subrotas obtido com os métodos GSProc e AgE. Neste conjunto, existe uma subrota inicial, S_2 , uma subrota final, S_1 , sendo as restantes intermédias e $P = 2$.

$$S_3 = \{(11,6); (6,4); (4,3); (3,6); (6,11)\}, \quad Q_3 = 3;$$

$$S_4 = \{(11,6); (6,5); (5,4); (4,5); (5,4); (4,6); (6,11)\}, \quad Q_4 = 4;$$

$$S_5 = \{(11,8); (8,9); (9,8); (8,11)\}, \quad Q_5 = 2;$$

$$S_6 = \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,11)\}, \quad Q_6 = 4.$$

Logo é necessário alterar o início de uma subrota e o fim de outra. Se, por exemplo, for considerada S_5 como subrota inicial e S_6 como subrota final, resultam as subrotas iniciais e finais:

$$S_2 = \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\},$$

$$S_5 = \{(1,8); (8,9); (9,8); (8,11)\},$$

$$S_1 = \{(11,3); (3,2); (2,1)\},$$

$$S_6 = \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,1)\},$$

mantendo-se as intermédias S_3 e S_4 . ◆

4.7. Análise Teórica da Heurística H1

Neste ponto apresenta-se um majorante para o desvio do erro relativo da heurística H1 relativamente ao valor do minorante. Relembre-se que H1 é composta pela sequência de métodos: GSCE, AgSeq e SolAdm. Como se verá, o cálculo deste desvio é possível por H1 se basear em GSCE, ou seja, em virtude do conjunto inicial de subrotas ser gerado da partição de um circuito Euleriano.

Seja:

- \underline{z} o minorante para o valor óptimo de PRRS, z^* , calculado pelo algoritmo LB descrito no capítulo 3.;
- \bar{z} o majorante para o valor óptimo de PRRS, z^* , calculado utilizando a heurística H1.

Defina-se:

- $lcl(v) = \text{Max}_{i \in V \setminus \{v\}} \{spl(i, v) + spl(v, i)\}, v \in V.$

- $gap = \frac{\bar{z} - \underline{z}}{\underline{z}}$ o desvio relativo entre \bar{z} e \underline{z} .

Considere-se, tal como no segundo capítulo, que os m arcos de procura estão ordenados por ordem não crescente das procuras, isto é, tais que: $q_1 \geq q_2 \geq \dots \geq q_m$.

Represente-se por ℓ o número máximo de arcos compatível com a capacidade de um veículo que, iniciando por servir o arco de maior procura associada, sirva os arcos pela ordem por que foram ordenados. Logo, ℓ é tal que:

$$\sum_{a=1}^{\ell} q_a \leq W < \sum_{a=1}^{\ell+1} q_a.$$

Seja $L = \left\lceil \frac{m}{\ell} \right\rceil$.

Começa por estabelecer-se um majorante para o desvio relativo do majorante para o valor óptimo obtido por H1 em relação ao minorante, para os casos em que apenas existe uma componente conexa em G_{LB} . O resultado será posteriormente estendido ao caso geral em que existem diversas componentes conexas.

Proposição 4.2: Considere-se um problema em que G_{LB} é formado por uma só componente conexa e tal que $P \geq \text{Max} \{d^-(I); d^+(I)\}$.

$$\text{Então, } \text{gap} \leq \frac{(L + P - 2) \text{ lcl}(n + 1)}{C_{A_R} + K \text{ scl}(n + 1)}. \quad (4.1)$$

Prova: Seja CE um circuito Euleriano no multigrafo associado à solução do algoritmo LB. Para gerar um conjunto de subrotas admissíveis, por GSCE, segue-se CE , a partir de $n + 1$, enquanto a capacidade dos veículos permite e enquanto não são encontrados os vértice l e $n + 1$.

Quando a capacidade dos veículos o exige, termina-se a subrota num vértice de CE e considera-se o caminho de menor custo em vazio desse vértice a $n + 1$ e inicia-se uma nova subrota considerando o caminho de menor custo em vazio de $n + 1$ até esse vértice. No pior caso, o vértice está o mais afastado possível de $n + 1$, sendo então percorrido um circuito de custo nunca superior a $\text{lcl}(n + 1)$. Se CE só fosse interrompido quando a capacidade dos veículos o exige seriam geradas, no máximo,

$$L = \left\lceil \frac{m}{\ell} \right\rceil$$

subrotas.

Porém CE é também interrompido se um dos vértices l ou $n + 1$ for encontrado o que se verifica, no máximo, $K + P$ vezes.

Assim, CE é interrompido $L + K + P$ vezes, no máximo.

Ao interromper as K vezes por encontrar $n + 1$, a sequência de arcos de CE é respeitada não existindo qualquer alteração de custos. O mesmo se verifica para pelo menos uma das L vezes e para pelo menos uma das P vezes. Para as restantes $(L + P - 2)$ vezes CE terá então que ser interrompido, no pior caso, no vértice a custo máximo de $n + 1$, sendo $\text{lcl}(n + 1)$ o custo associado.

No pior caso não é possível a agregação de subrotas na segunda fase da heurística, permanecendo assim inalterado o valor da solução gerada.

Sendo, por hipótese, $P \geq \text{Max} \{d^-(I); d^+(I)\}$ existem no início da terceira fase da heurística não mais de P subrotas iniciais e de P subrotas finais. Assim, nesta fase nunca se verificarão, em nenhuma das subrotas, trocas de vértices finais nem iniciais que possam conduzir a um aumento no valor da solução.

Então,

$$\bar{z} \leq \underline{z} + (L + P - 2) \text{lcl}(n + I) \Rightarrow \\ \Rightarrow \frac{\bar{z} - \underline{z}}{\underline{z}} \leq \frac{(L + P - 2) \text{lcl}(n + I)}{\underline{z}}.$$

Sendo, $\underline{z} \geq C_{A_R} + K \text{scl}(n + I)$,

$$\text{vem: } \text{gap} \leq \frac{(L + P - 2) \text{lcl}(n + I)}{C_{A_R} + K \text{scl}(n + I)}. \quad \square$$

Caso $P < \text{Max} \{d^-(I); d^+(I)\}$ então P deve ser substituído por

$$\bar{d} = \text{Max} \{d^-(I); d^+(I)\}$$

na fórmula (4.1). Deve ainda ser somado ao numerador o custo de ter que substituir, no pior caso, $(\bar{d} - P)$ subrotas iniciais por intermédias e o de ter que substituir $(\bar{d} - P)$ subrotas finais por intermédias, dado, no pior caso, por $(\bar{d} - P)(\text{spl}(1, n + I) + \text{spl}(n + I, I))$. Logo é ainda possível estabelecer um majorante para o maior desvio relativo quando $P < \text{Max} \{d^-(I); d^+(I)\}$.

O resultado da proposição 4.2 pode generalizar-se para o caso de existirem nc componentes conexas, como se estabelece de seguida.

Corolário 4.1: Considere-se que em G_{LB} existem nc componentes conexas.

Para a i -ésima componente conexa, seja:

- m_i o número de arcos de procura na componente;
- ℓ_i o número máximo de arcos na componente, compatível com a capacidade de um veículo que iniciando por servir o arco de maior procura associada, sirva os arcos por ordem decrescente de procuras;

$$\bullet L_i = \left\lceil \frac{m_i}{\ell_i} \right\rceil;$$

$$\bullet L = \sum_{i=1}^{nc} L_i.$$

Então,

$$gap \leq \frac{(L + P - 2) lcl(n+1)}{C_{A_R} + K scl(n+1)}. \quad (4.2)$$

Prova: Da proposição 4.2, considerando cada componente conexa como um grafo e definindo:

$$lcl^i(n+1) = \text{Max}_{k \in V_R^i} \{ spl(k, n+1) + spl(n+1, k) \},$$

Vem $lcl^i(n+1) \leq lcl(n+1)$ e

$$gap \leq \frac{\sum_{i=1}^{nc} [L_i lcl^i(n+1)] + (P-2) lcl(n+1)}{\underline{z}}$$

$$gap \leq \frac{\left[\sum_{i=1}^{nc} L_i + P - 2 \right] lcl(n+1)}{\underline{z}} \Leftrightarrow$$

$$\Leftrightarrow gap \leq \frac{[L + P - 2] lcl(n+1)}{\underline{z}}$$

$$\Leftrightarrow gap \leq \frac{(L + P - 2) lcl(n+1)}{C_{A_R} + K scl(n+1)}. \quad \square$$

4.8. *Método Melhorativo*

Os métodos melhorativos, tal como o nome indica, têm como principal objectivo melhorar soluções admissíveis. Em cada iteração tenta então obter-se uma solução admissível melhor, ou seja, de menor valor, que a actual solução. Torna-se assim imprescindível a existência de uma solução admissível inicial, que pode ser obtida pelos métodos propostos no capítulo anterior. Este tipo de métodos foram inicialmente propostos na década de sessenta, por Lin e Kernighan ([59], [60]) para o problema do caixeiro viajante.

4.8.1. *Introdução*

Dada uma solução admissível pode tentar obter-se outra solução admissível de menor valor, admitindo trocas de arcos nas subrotas. A redução no custo da solução está obviamente associada a uma eventual diminuição no custo dos trajectos em vazio na solução de partida.

Uma forma de tentar atingir este objectivo consiste em, se possível, efectuar trocas entre arcos que apareçam replicados numa ou em diferentes subrotas. De forma a não obter um método muito exaustivo e, conseqüentemente, muito moroso, apenas se admitiram trocas entre o serviço de arcos e suas eventuais travessias em vazio. Assim, tenta trocar-se a posição em que actualmente é servido cada um dos arcos que sendo de procura surjam replicados, no conjunto de rotas admissíveis em presença.

No primeiro tipo de troca que se apresenta tentam melhorar-se as actuais subrotas, uma a uma, pela eliminação de repetições de arcos, em cada subrota. Para tal analisam-se, por exemplo, as soluções associadas à troca do serviço de um arco que numa mesma subrota seja servido e percorrido em vazio. Assim, troca-se, se for vantajoso, a posição de cada arco que, numa mesma subrota, seja servido e atravessado em vazio, passando este a ser percorrido em serviço quando era inicialmente percorrido em vazio e vice-versa.

Melhoradas, se possível, as actuais subrotas individualmente, tenta melhorar-se o conjunto das subrotas. Estudam-se então os casos em que é possível e vantajosa a troca de arcos entre subrotas distintas. Considerado um arco que apareça repetido em duas ou mais subrotas, sendo servido numa delas, tenta trocar-se o seu serviço de subrota. Ou seja, analisam-se os casos em que um arco

pode e deve passar a ser servido numa subrota em que era percorrido em vazio e passar a ser percorrido em vazio na subrota em que era inicialmente servido.

Tal como anteriormente, um arco percorrido em vazio representa o caminho de menor custo entre os respectivos vértices, no grafo inicial. Como justificado, tal representação permite uma implementação mais eficiente embora possa impossibilitar, por vezes, a detecção de subrotas adjacentes. Assim, os arcos que antecedem e sucedem, em qualquer subrota, um arco de procura nula (representativo de um caminho de menor custo) são arcos servidos nessa subrota.

Neste método, um arco (x, y) é representado por

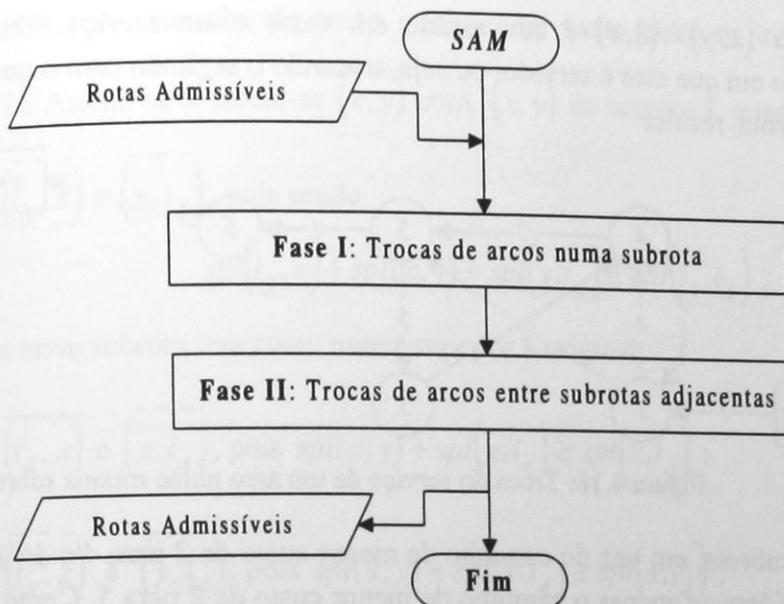
- $\overline{(x, y)}$ sempre que se pretenda frisar que este está a ser percorrido em serviço;
- $\overline{\overline{(x, y)}}$ se estiver a ser percorrido em vazio, e isso seja relevante para o que se pretende.

4.8.2. Descrição do Método

Introdução

O método melhorativo SAM ⁽¹⁾ pode ser considerado um método de duas fases, sendo que em cada uma se tenta melhorar uma solução admissível existente. Assim, na primeira fase tentam-se melhorar as subrotas uma a uma, admitindo trocas de arcos numa mesma subrota. Na segunda fase tenta-se melhorar o conjunto de rotas admissíveis, considerando certas trocas entre subrotas adjacentes. No fluxograma 4.3 descreve-se o método geral, sendo nos pontos seguintes justificadas cada uma das suas duas fases, com mais detalhe.

⁽¹⁾ SAM – Solução Admissível Melhor.



Fluxograma 4.3: Método melhorativo – SAM

Trocas de Arcos numa Subrota

Neste tipo de trocas pretendem eliminar-se repetições de arcos em cada subrota. Para tal, analisa-se em que casos se deve alterar a posição em que um arco é servido, numa subrota em que esse arco aparece repetido.

Considere-se a subrota:

$$S = \left\{ (i_1, x); (\overline{x}, \overline{y}); (y, i_2); (i_2, i_3); (i_3, x); (\overline{x}, y); (y, i_4); (i_4, i_5); (i_5, i_6) \right\}$$

em que (x, y) aparece repetido, sendo servido numa das vezes, como se exemplifica de seguida.

Exemplo 4.3: Considere-se a subrota admissível representada na figura seguinte, onde os números sobre os arcos indicam a sequência na subrota

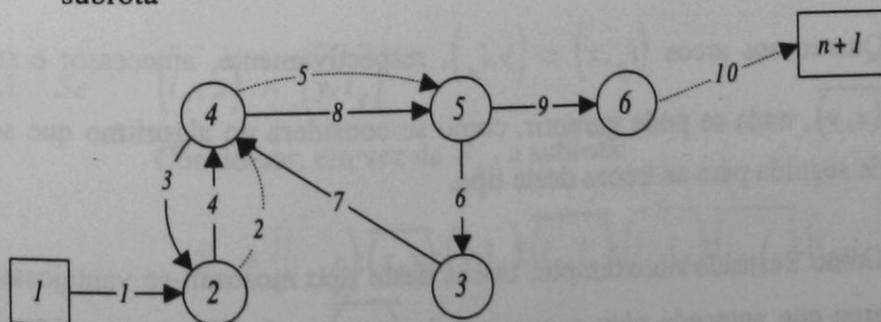


Figura 4.17: Subrota com réplicas de arcos servidos

O arco $(x, y) = (2, 4)$ é atravessado em vazio e em serviço. Trocando a posição em que este é servido, ou seja, trocando o segundo com o quarto arco na subrota, resulta

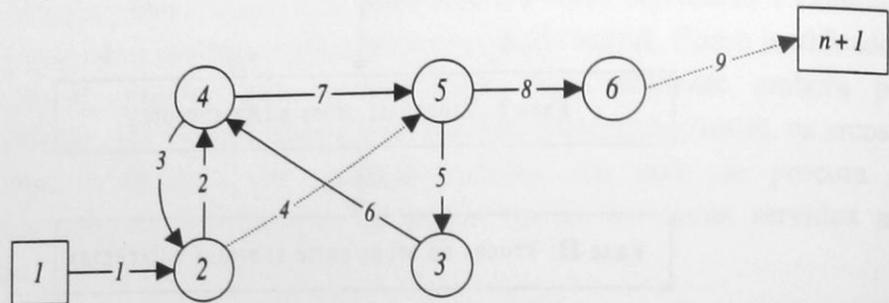


Figura 4.18: Troca do serviço de um arco numa mesma subrota

Nesta subrota, em vez do caminho de menor custo de 2 para 4 e do de 4 para 5, considera-se apenas o caminho de menor custo de 2 para 5. Como é válida a desigualdade

$$spl(2,4) + spl(4,5) \geq spl(2,5).$$

a subrota resultante da troca tem custo total nunca superior ao da subrota inicial.

Note-se que esta troca só é eventualmente vantajosa caso a troca de serviço do arco que é servido e percorrido em vazio faça com que passem a existir pelo menos dois arcos em vazio consecutivos, na sequência de arcos considerada na subrota. De facto, se for trocado o serviço do arco $(4,5)$ de posição (figura 4.17), ou seja, trocar no conjunto inicial o quinto com o oitavo arcos, não altera o valor da solução. ♦

Tendo em conta a representação adoptada, em que um caminho em vazio de i para j , no grafo original, é substituído pelo arco (\bar{i}, \bar{j}) , torna-se óbvio que os arcos que antecedem e sucedem na subrota são arcos percorridos em serviço, ou seja, (\bar{i}_1, x) e (y, \bar{i}_2) .

Quanto aos arcos (i_3, x) e (y, i_4) , respectivamente, antecessor e sucessor de (x, y) , nada se pode garantir, como se considera no algoritmo que se descreve de seguida para as trocas deste tipo.

Como ilustrado no exemplo, trocas deste tipo mostram-se vantajosas quando o arco que antecede e/ou o que sucede (x, y) estão a ser percorridos em vazio, originando uma sequência de, pelo menos, dois caminhos em vazio que serão

substituídos pelo caminho de custo mínimo em vazio entre os extremos da sequência. Assim, deve trocar-se $\overline{(x, y)}$ com $\overline{(x, y)}$ na subrota S , quando

- $\overline{(i_3, x)}$ e $\overline{(y, i_4)}$, pois sendo

$$spl(i_3, x) + spl(x, y) + spl(y, i_4) \geq spl(i_3, i_4),$$

a nova subrota tem custo nunca superior à original.

- $\overline{(i_3, x)}$ e $\overline{(y, i_4)}$, pois $spl(x, y) + spl(y, i_4) \geq spl(x, i_4)$;

- $\overline{(i_3, x)}$ e $\overline{(y, i_4)}$, pois $spl(x, y) + spl(y, i_4) \geq spl(x, i_4)$.

Algoritmo Trocas1: Troca do Serviço de Arcos numa mesma Subrota

0. Dados: Conjunto de ns subrotas admissíveis;

1. Fazer $k \leftarrow 1$;

2. Se não existem arcos servidos em S_k e percorridos em vazio

Fazer $k \leftarrow k + 1$

Se $k \leq ns$ Voltar a 2.

C.c., Fim;

C.c., Ir para 3.

3. Seja:

$$S_k = \left\{ (i_1, x); \overline{(x, y)}; (y, i_2); (i_2, i_3); (i_3, x); \overline{(x, y)}; (y, i_4); (i_4, i_5); (i_5, i_6) \right\}$$

3.1 Se $\overline{(i_3, x)} \wedge \overline{(y, i_4)}$

Considerar, em vez de S_k , a subrota:

$$\tilde{S}_k = \left\{ (i_1, x); \overline{(x, y)}; (y, i_2); (i_2, i_3); \overline{(i_3, i_4)}; \overline{(i_4, i_5)}; (i_5, i_6) \right\}$$

$$3.2 \text{ Se } \overline{(i_3, x)} \wedge \overline{\overline{\overline{(y, i_4)}}}$$

Considerar, em vez de S_k , a subrota:

$$\tilde{S}_k = \left\{ \overline{(i_1, x)}; \overline{(x, y)}; \overline{(y, i_2)}; \overline{(i_2, i_3)}; \overline{(i_3, x)}; \overline{\overline{\overline{(x, i_4)}}}; \overline{(i_4, i_5)}; \overline{(i_5, i_6)} \right\},$$

$$3.3 \text{ Se } \overline{\overline{\overline{(i_3, x)}}} \wedge \overline{(y, i_4)}$$

Considerar, em vez de S_k , a subrota:

$$\tilde{S}_k = \left\{ \overline{(i_1, x)}; \overline{(x, y)}; \overline{(y, i_2)}; \overline{(i_2, i_3)}; \overline{\overline{\overline{(i_3, y)}}}; \overline{\overline{\overline{(y, i_4)}}}; \overline{(i_4, i_5)}; \overline{(i_5, i_6)} \right\}$$

$$3.4 \text{ Se } \overline{(i_3, x)} \wedge \overline{\overline{\overline{(y, i_4)}}} \text{ Voltar a 2.}$$

Logo, se um arco de procura aparece repetido na subrota em que é servido, torna-se vantajoso trocar a posição do seu serviço se pelo menos um dos arcos que liga com o servido, na subrota inicial, for percorrido em vazio. Assim, mal o serviço de um arco é trocado de posição o mesmo arco sai de análise. Ou seja, se aparece repetido mais de uma vez só se consegue trocar uma vez no máximo. Pois, quando é efectuada uma troca o arco percorrido em serviço passa a ficar "ligado" a arcos que também são servidos, não sendo portanto vantajoso admitir novas trocas para esse mesmo arco.

Se existir mais de um arco repetido numa mesma subrota faz-se esta análise sequencialmente seguindo a ordem com que estes aparecem na subrota. Tenta-se assim não tornar muito pesadas, em termos de tempos de execução, estas análises.

Trocas de Arcos entre Subrotas Adjacentes

Na segunda fase do método melhorativo SAM pretende, se possível, melhorar-se o actual conjunto de subrotas. Para tal, vão identificar-se os casos em que é possível e vantajosa a troca do serviço de um arco entre duas subrotas.

Seja então (x, y) um arco que aparece repetido em duas subrotas, sendo servido numa delas. Pretende-se saber se, sendo possível, haverá vantagens em trocar o seu serviço de subrota. Isto é, em passar a servi-lo na subrota em que ele é percorrido em vazio, caso a capacidade do veículo o permita, e passar a ser percorrido em vazio na subrota em que é actualmente servido.

Considerem-se então as duas subrotas distintas:

$$S_i = \left\{ (i_1, i_2); (i_2, i_3); (i_3, x); \overline{(x, y)}; (y, i_4); (i_4, i_5); (i_5, i_6) \right\}$$

$$S_j = \left\{ (j_1, j_2); (j_2, x); \overline{(x, y)}; (y, j_3); (j_3, j_4); (j_4, j_5) \right\}.$$

Note-se que, dada a representação adoptada, os arcos que antecedem e sucedem

$\overline{(x, y)}$, em S_i , são percorridos em serviço, ou seja, (i_3, x) e (y, i_4) .

Exemplo 4.4: Considerem-se as subrotas admissíveis representadas na figura seguinte, onde o primeiro número sobre cada arco identifica a subrota e o segundo o número do arco na sequência considerada para formar a respectiva subrota.

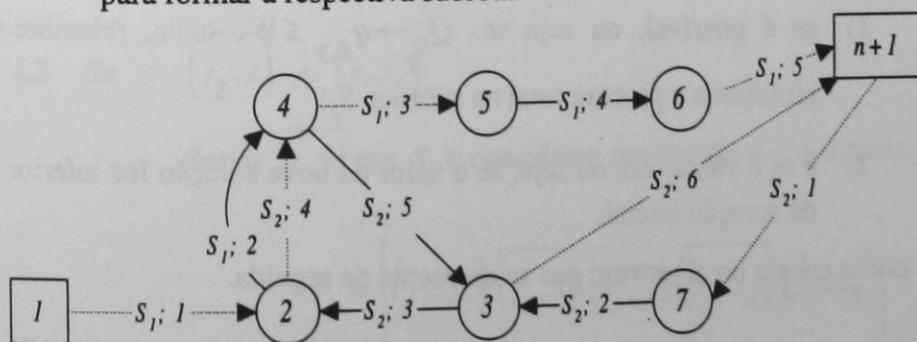


Figura 4.19: Subrotas adjacentes com duplicação de arcos

O arco $(x, y) = (2, 4)$ é atravessado em serviço na subrota S_1 e em vazio na subrota S_2 . Trocando, se compatível com a capacidade dos veículos, o serviço de $(2, 4)$ entre estas duas subrotas, resultaria:

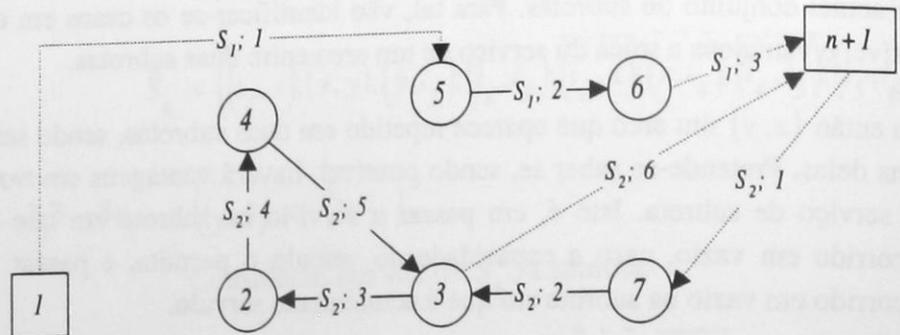


Figura 4.20: Troca do serviço de um arco entre subrotas adjacentes

Nestas duas subrotas, em vez dos caminhos de menor custo de 1 para 2, de 2 para 4 e de 4 para 5, considera-se apenas o caminho de menor custo de 1 para 5, pois é válida a desigualdade:

$$spl(1,2) + spl(2,4) + spl(4,5) \geq spl(1,5).$$

Como no caso anterior, esta troca só é eventualmente vantajosa caso o arco em questão seja precedido e/ou sucedido por arcos percorridos em vazio na subrota em que é servido. ♦

Então, só se passa a servir o arco (x, y) em S_i , resultando

$$\tilde{S}_i = \{ (i_1, i_2); (i_2, i_3); \overline{(i_3, x)}; \overline{(x, y)}; \overline{(y, i_4)}; (i_4, i_5); (i_5, i_6) \}$$

- 1) se é possível, ou seja se: $Q_{S_i} + q_{x,y} \leq W$, onde, relembre-se, Q_{S_i} representa a procura total na subrota S_i ;
- 2) e se é vantajoso, ou seja, se o valor da nova solução for inferior ao valor da solução inicial,

como consta no algoritmo que se apresenta de seguida.

Algoritmo Trocas2: Troca do Serviço de Arcos entre Subrotas

0. Dados: Conjunto de ns subrotas admissíveis;
1. Fazer $i \leftarrow 1$; $j \leftarrow 1$;
2. Fazer $j \leftarrow j + 1$;

3. Se $j \leq ns$

Se S_i e S_j têm um arco (x, y) em comum, sendo percorrido em serviço em S_j ⁽¹⁾ e $(Q_{S_i} + q_{x,y} \leq W)$

Ir para 4.

C.c., Voltar a 2.;

C.c., Fazer $i \leftarrow i+1$; $j \leftarrow i+1$;

Se $i < ns$ Voltar a 3.

C.c., Fim;

4. Sejam:

$$S_i = \left\{ (i_1, i_2); (i_2, i_3); \overline{(i_3, x)}; \overline{(x, y)}; \overline{(y, i_4)}; (i_4, i_5); (i_5, i_6) \right\}$$

$$S_j = \left\{ (j_1, j_2); (j_2, x); \overline{(x, y)}; \overline{(y, j_3)}; (j_3, j_4); (j_4, j_5) \right\}.$$

4.1 Se $\overline{(j_2, x)} \wedge \overline{(y, j_3)}$

Servir $\overline{(x, y)}$ em S_i e considerar, em vez de S_j , a subrota:

$$\tilde{S}_j = \left\{ \overline{(j_1, j_2)}; \overline{(j_2, j_3)}; \overline{(j_3, j_4)}; (j_4, j_5) \right\};$$

4.2 Se $\overline{(j_2, x)} \wedge \overline{(y, j_3)}$

Servir $\overline{(x, y)}$ em S_i e considerar, em vez de S_j , a subrota:

$$\tilde{S}_j = \left\{ (j_1, j_2); \overline{(j_2, x)}; \overline{(x, j_3)}; \overline{(j_3, j_4)}; (j_4, j_5) \right\};$$

(1) Considera-se, por simplificação, que na solução actual o arco está a ser percorrido em serviço em S_j . Obviamente que ser percorrido em serviço em S_i e atravessado em vazio em S_j proporciona um estudo semelhante.

$$4.3 \text{ Se } (\overline{j_2, x}) \wedge (\overline{y, j_3})$$

Servir $(\overline{x, y})$ em S_i e considerar, em vez de S_j , a subrota:

$$\tilde{S}_j = \left\{ (\overline{j_1, j_2}); (\overline{j_2, y}); (\overline{y, j_3}); (\overline{j_3, j_4}); (\overline{j_4, j_5}) \right\};$$

$$4.4 \text{ Se } (\overline{j_2, x}) \wedge (\overline{y, j_3}) \text{ Voltar a 3..}$$

Logo, se um arco de procura aparece repetido em duas subrotas, torna-se vantajoso trocar o seu serviço de subrota se pelo menos um dos arcos que liga com o servido for percorrido em vazio.

Assim, tal como anteriormente, mal o serviço de um arco é trocado de posição ou de subrota o arco sai de análise. Pois, mesmo que apareça repetido mais de uma vez só se consegue trocar, o seu serviço, uma vez no máximo, dado ficar "ligado" a arcos que são percorridos em serviço.

Se existirem dois ou mais arcos repetidos, as trocas vão sendo consideradas sequencialmente, pela ordem por que as subrotas foram construídas. Mais uma vez, tenta-se assim não sobrecarregar os algoritmos com este tipo de análise.

4.8.3. Exemplo A

Retome-se, para o exemplo protótipo que se tem vindo a apresentar, a solução admissível obtida pela sequência de algoritmos GSProc, AgE e SolAdm, ou seja:

$$\begin{aligned}
 S_1 &= \{(11,3); (3,2); (2,1)\}, & Q_1 &= 2; \\
 S_2 &= \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\}, & Q_2 &= 4; \\
 S_3 &= \{(11,6); (6,4); (4,3); (3,6); (6,11)\}, & Q_3 &= 3; \\
 S_4 &= \{(11,6); (6,5); (5,4); (4,5); (5,4); (4,6); (6,11)\}, & Q_4 &= 4; \\
 S_5 &= \{(1,8); (8,9); (9,8); (8,11)\}, & Q_5 &= 2; \\
 S_6 &= \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,1)\}, & Q_6 &= 4.
 \end{aligned}$$

Apenas na subrota S_4 existe réplica de um arco servido, o arco $(5,4)$. Porém, este arco é sempre antecedido e sucedido, em S_4 , por arcos percorridos em serviço (figura 4.21). Logo, não se consideram trocas no serviço deste arco.

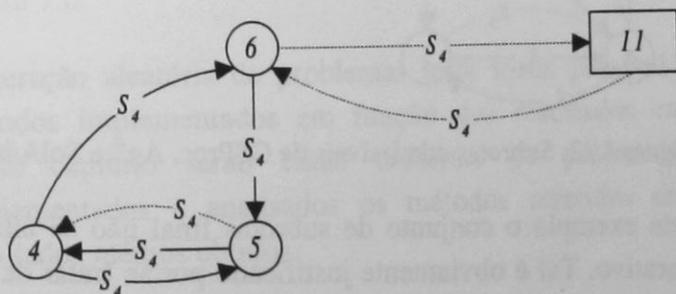


Figura 4.21: Subrotas admissíveis de GSProc, AgE e SolAdm

É de realçar que neste caso não existem, em nenhuma subrota, arcos que sendo de procura sejam percorridos em vazio noutra subrota que não aquela em que são servidos. Logo, não há lugar a trocas de serviço entre subrotas adjacentes.

Considere-se agora o conjunto de subrotas obtido com a sequência de métodos GSProc, AgS e SolAdm, sobre o mesmo exemplo, ou seja:

$$\begin{aligned}
 S_1 &= \{(11,3); (3,2); (2,1)\}, & Q_1 &= 2; \\
 S_2 &= \{(1,3); (3,4); (4,7); (7,5); (5,7); (7,11)\}, & Q_2 &= 4;
 \end{aligned}$$

$$\begin{aligned}
 S_3 &= \{(11,6); (6,4); (4,5); (5,4); (4,3); (3,6); (6,11)\}, & Q_3 &= 4; \\
 S_4 &= \{(11,6); (6,5); (5,4); (4,6); (6,11)\}, & Q_4 &= 3; \\
 S_5 &= \{(1,8); (8,9); (9,8); (8,11)\}, & Q_5 &= 2; \\
 S_6 &= \{(11,8); (8,10); (10,9); (9,10); (10,8); (8,1)\}, & Q_6 &= 4.
 \end{aligned}$$

Neste caso poder-se-ia pensar em trocar o serviço do arco $(5,4)$ da subrota S_4 para a subrota S_3 . Mesmo que fosse possível, o que não se verifica dado os veículos terem capacidade $W = 4$, igual à procura total de S_3 , não há vantagens, pois tanto o arco que precede $(5,4)$, em S_4 , como o que o sucede são arcos percorridos em serviço (figura 4.22). Assim, tal só significaria que $(5,4)$ seria percorrido em vazio em S_4 em vez de o ser em S_3 , não alterando o custo total da solução.

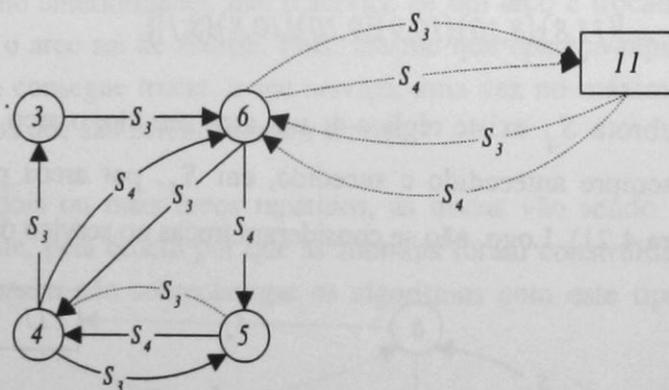


Figura 4.22: Subrotas admissíveis de GSProc, AgS e SolAdm

Logo, para este exemplo o conjunto de subrotas final não se altera perante o método melhorativo. Tal é obviamente justificado por se tratar de um exemplo de pequenas dimensões, onde são “poucos” os arcos percorridos em vazio nas soluções admissíveis obtidas com os métodos do capítulo anterior. ◆

Em suma, espera-se que este método seja tanto mais efectivo, proporcionando melhorias nas soluções admissíveis, quanto maiores forem as dimensões dos problemas. Pois será nestes casos que as soluções admissíveis obtidas com os métodos do capítulo anterior proporcionarão soluções com um maior número de arcos repetidos.

Capítulo 5. Resultados Computacionais

5.1. Introdução

Os três métodos heurísticos H1, H2 e H3 e o método para determinar minorantes, LB, foram codificados em Pascal e compilados em Turbo Pascal 7.0.

A geração aleatória de problemas teste torna possível a análise dos métodos implementados em função dos resultados computacionais. Neste capítulo serão então descritos os processos de geração implementados e analisados os métodos referidos em função dos resultados médios obtidos.

Os problemas teste, gerados aleatoriamente, podem subdividir-se em dois tipos, como se apresentará. Para o primeiro tipo, T1, foram geradas redes pouco densas, ou seja, redes em que o número total de arcos de procura seja relativamente "pequeno". No segundo tipo de problemas, T2, pretende obter-se redes planares e menos densas que as de tipo T1, designadas por isso por redes esparsas. Tenta-se assim a geração de redes que possam ser consideradas similares a redes de estradas.

No ponto seguinte apresentam-se e analisam-se os resultados para os problemas de tipo T1, de mais fácil geração. O terceiro ponto deste capítulo é então dedicado a problemas de tipo T2. A justificação dos métodos implementados na geração de problemas deste tipo é seguida da apresentação e análise dos respectivos resultados computacionais.

Como inicialmente referido, todos os programas foram escritos em Pascal, tendo sido utilizado o compilador Turbo Pascal 7.0. Este compilador impõe certas limitações quer às dimensões dos problemas teste, quer ao tipo de dados. Não foram por isso gerados problemas com mais de 50 vértices nem com valores para os parâmetros muito elevados, tendo sido, no geral, as variáveis declaradas como variáveis inteiras.

Os resultados foram todos obtidos num processador *Pentium* a 100MHz, configurado com 24MB de memória.

O desvio relativo do valor de uma solução admissível, \bar{z} , em função do valor do minorante obtido pelo método LB, \underline{z} , é calculado, em percentagem, pela expressão

$$gap = \frac{\bar{z} - \underline{z}}{\underline{z}} \times 100, \quad (5.1)$$

já anteriormente referida.

5.2. Problemas Teste em Redes Pouco Densas

5.2.1. Geração

O primeiro tipo de dados considerado foi obtido pela geração aleatória de redes "pouco densas" e conexas. As procuras foram geradas de acordo com a distribuição uniforme $U(6, 20)$. Os custos em vazio associados a cada arco foram gerados pela $U(2, 15)$, para os arcos não incidentes em $n + 1$, enquanto para os arcos incidentes em $n + 1$ foi utilizada a $U(30, 40)$.

Os custos de percorrer cada arco em serviço foram calculados por

$$c_{i,j} = 0.5 d_{i,j} q_{i,j}, \quad \forall (i,j) \in A_R,$$

resultando $0.5 q_{i,j} > 1 \Rightarrow c_{i,j} > d_{i,j}, \quad \forall (i,j) \in A_R$, conforme pretendido.

Tendo sido testados problemas com um máximo de 50 vértices, considerou-se igual a dois o número de veículos. Pois, se por um lado há que admitir mais de um veículo por estar em estudo o caso múltiplo, por outro, não faz sentido, em problemas pouco densos destas dimensões admitir a existência de um elevado número de veículos. Pensa-se nestes casos ser mais relevante, em termos de análise de resultados, a existência de diversas subrotas que de diversos veículos.

Para um mesmo número de vértices foram considerados veículos de diferentes capacidades, estando os valores considerados indicados nas tabelas de resultados computacionais detalhadas em anexo. Foram também testados problemas de diferentes densidades (assumindo valores entre os 30% e os 40%), definidas pelo quociente entre o número de arcos de procura considerados e o número total de possíveis arcos de procura, ou seja:

$$densidade = \frac{|A_R|}{n(n-1)}.$$

Cada problema teste representa valores médios de dez problemas gerados aleatoriamente com as mesmas características, ou seja, médias para problemas que tendo os mesmos dados iniciais, se distinguem entre si dada a geração aleatória de alguns dos parâmetros envolvidos.

Para este tipo de problemas apenas se apresentam e analisam os resultados obtidos pela heurística H1 e, obviamente, por LB uma vez que o "gap"

calculado em função dos valores da heurística e do minorante, só assumirá valores pequenos caso ambos os métodos sejam "bons".

Nesta tese não se apresentam resultados dos restantes métodos em problemas T1, por H1 ter proporcionado resultados muito bons. Assim, sabendo de início que H1 é das heurísticas a que necessita de menor tempo de execução, pensa-se que não se justifica o estudo comparativo entre métodos alternativos em problemas T1. Em problemas deste tipo a heurística H1 fornece, como se verá, soluções admissíveis bastante razoáveis, quer em termos da configuração de subrotas, quer em termos do valor da solução admissível.

As restantes heurísticas serão comparadas com a primeira em problemas que proporcionam piores resultados médios, como é o caso dos problemas de tipo T2 gerados aleatoriamente.

5.2.2. Resultados Computacionais

Os gráficos seguintes sintetizam os resultados computacionais mais detalhados que se podem consultar no anexo A3. Assim, primeiro apresentam-se e analisam-se os gráficos associados aos valores dos desvios percentuais relativos calculados, "gap".

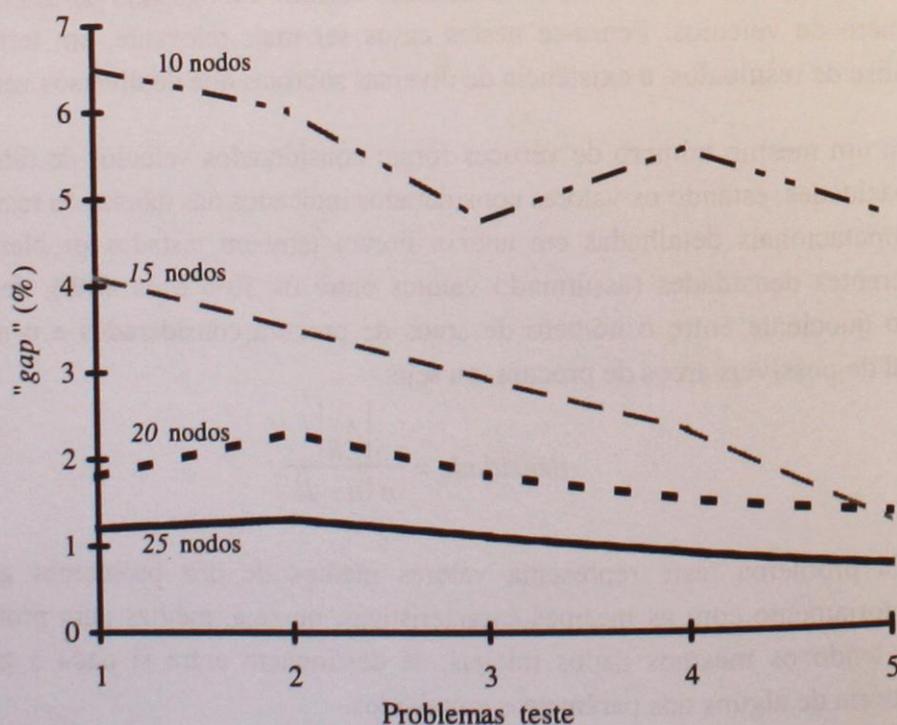


Figura 5.1: Desvios relativos de H1 em problemas T1, com 10 a 25 vértices

É notório quer no gráfico da figura 5.1, quer no da figura 5.2, uma diminuição no valor do "gap" com o aumento das dimensões dos problemas, ou seja do número de nodos e , e, conseqüentemente, do número de arcos. Note-se que para problemas com 10 a 25 vértices (figura 5.1) os valores percentuais do "gap" médio variam entre 0.5% e 7%, estando a curva associada aos problemas com 10 vértices sempre acima das restantes. Para os problemas de maiores dimensões os valores percentuais do "gap" médio variam entre 0.15% e 1.2%, sendo os mais elevados atingidos para os problemas de 30 vértices e os mais baixos para os problemas com 50 vértices.

Relativamente a redes de iguais dimensões, ou seja com o mesmo número de vértices, nota-se uma tendência de diminuição nos valores do "gap" com o aumento da capacidade dos veículos, sabendo que os problemas foram numerados por ordem crescente de W . Tal estará obviamente relacionado com a facto de veículos de maiores capacidades proporcionarem a geração de soluções admissíveis com um menor número de subrotas.

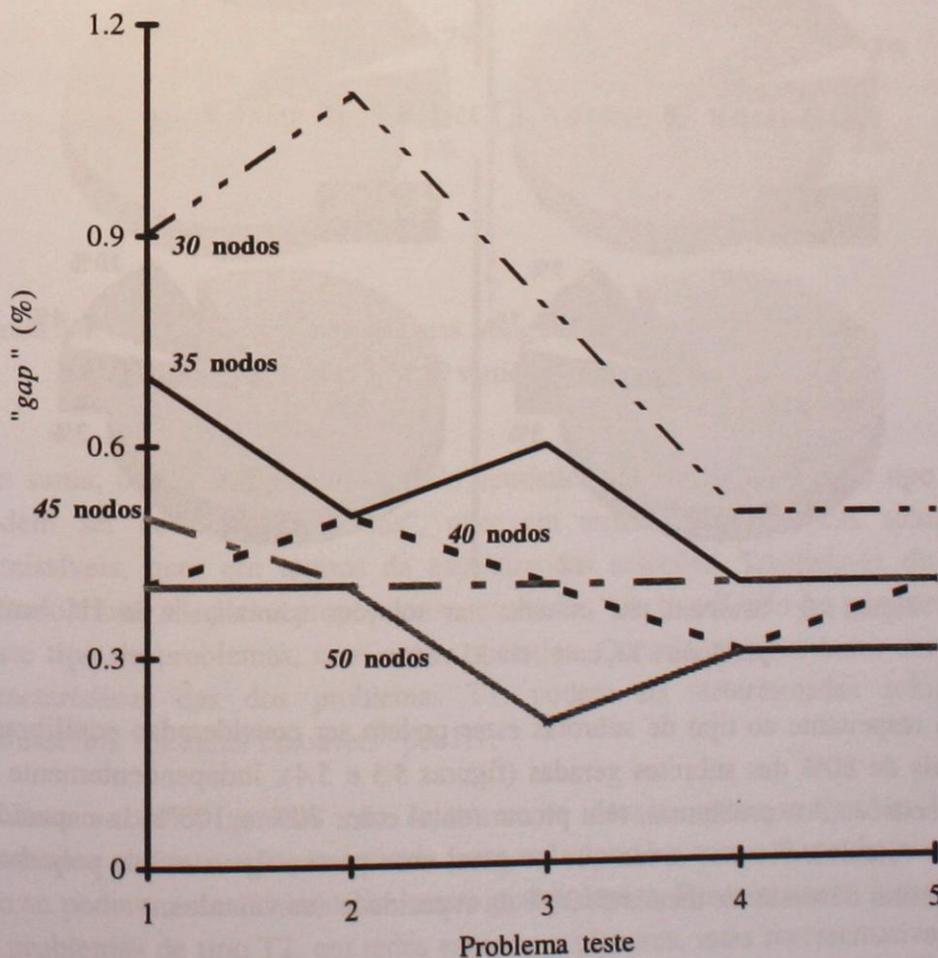


Figura 5.2: Desvios relativos de H1 em problemas T1, com 30 a 50 vértices

Apresentados e analisados os valores dos desvios relativos das soluções admissíveis proporcionadas por H1, passa-se agora à análise das respectivas soluções. Esta análise é feita em função do (des)equilíbrio das subrotas geradas.

As figuras 5.3 e 5.4 representam os resultados médios para cada uma das dimensões testadas. Sendo os valores para cada um dos problemas teste similares aos valores médios, como se pode confirmar em anexo, a análise dos resultados pode ser feita globalmente.

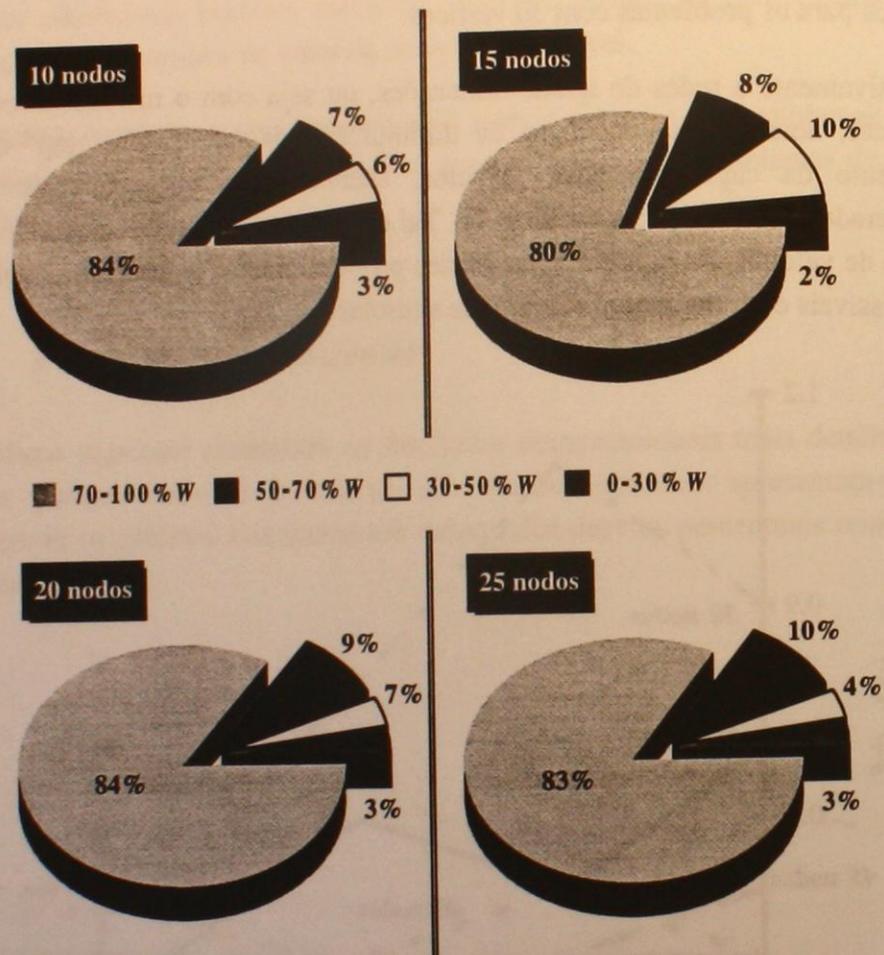


Figura 5.3: Estrutura das subrotas nas soluções admissíveis de H1 em problemas T1, com 10 a 25 vértices

No respeitante ao tipo de subrotas estas podem ser consideradas equilibradas. Mais de 80% das subrotas geradas (figuras 5.3 e 5.4), independentemente das dimensões dos problemas, têm procura total entre 70% e 100% da capacidade dos veículos. Note-se ainda que no geral uma percentagem muito pequena de subrotas necessita de menos de 30% da capacidade dos veículos.

Obviamente que os valores do "gap" têm que estar de algum modo relacionados com o tipo de subrotas que se obtêm. Sendo estes tão pequenos, seria de esperar

que o tipo de soluções admissíveis, em função da estrutura das subrotas, fosse “razoável”.

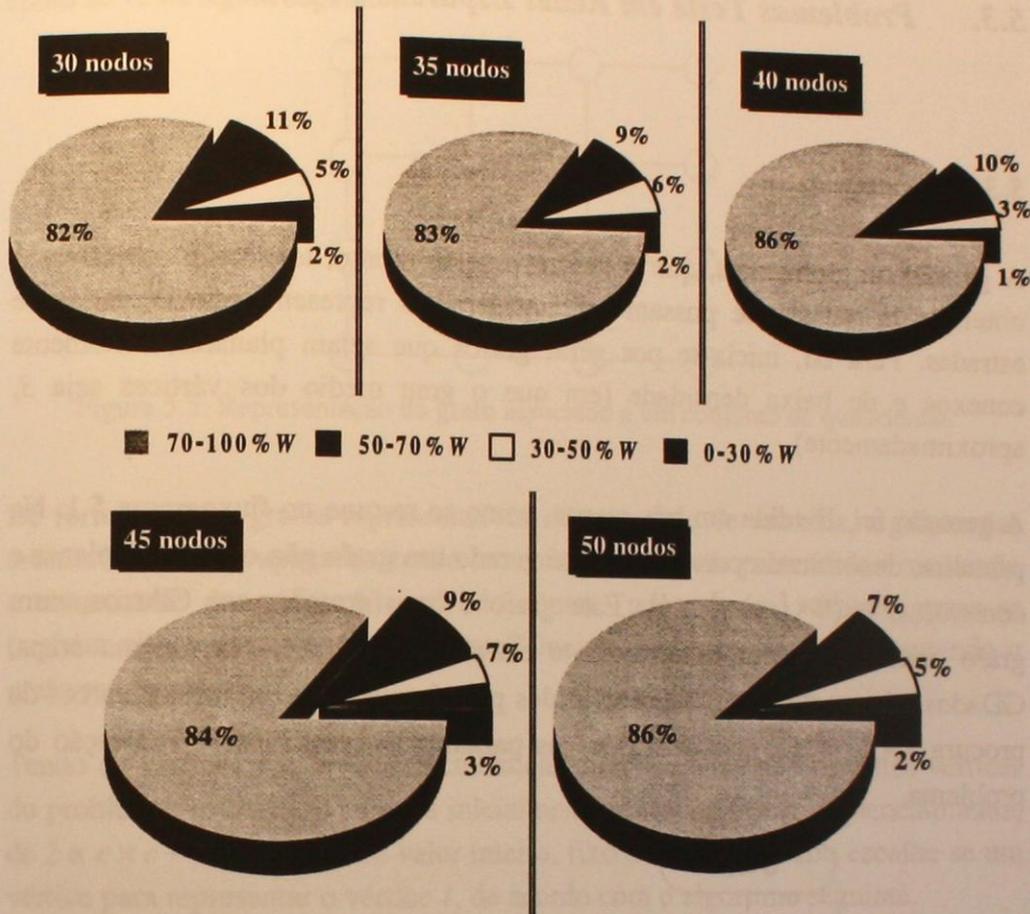


Figura 5.4: Estrutura das subrotas nas soluções admissíveis de H1 em problemas T1, com 30 a 50 vértices

Em suma, os resultados obtidos pela heurística H1 em problemas de tipo T1, podem ser considerados “bons”, quer em termos do valor das soluções admissíveis, quer em termos da estrutura das soluções. Tratando-se de um método muito simples a proporcionar resultados “tão bons”, não foi comparado, neste tipo de problemas, com outros métodos. Ou seja, perante redes com as características das dos problemas T1, podem ser determinadas soluções admissíveis “bastante razoáveis” por H1.

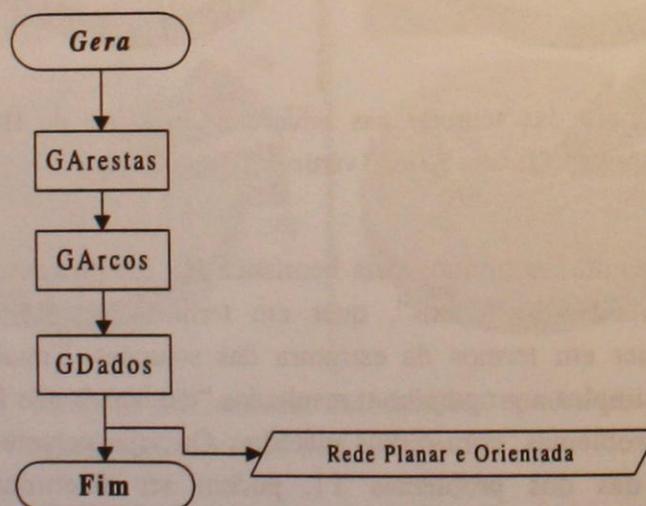
Embora as redes geradas não sejam muito densas, pensa-se que os bons resultados obtidos estão, em grande parte, relacionados com o facto de também não se poderem considerar redes muito pouco densas. Passa-se assim à geração de problemas de tipo T2, em redes esparsas e planares, mais representativas de redes de estradas.

5.3. Problemas Teste em Redes Esparsas e Planares

5.3.1. Geração

A geração de problemas, que se descreve neste ponto, teve como objectivo a obtenção de redes que possam ser consideradas representativas de redes de estradas. Para tal, inicia-se por gerar grafos que sejam planares, fortemente conexos e de baixa densidade (em que o grau médio dos vértices seja 3, aproximadamente).

A geração foi dividida em três etapas, como se resume no fluxograma 5.1. Na primeira, denominada por GArestas, é gerado um grafo não orientado, planar e conexo, $G' = (V \setminus \{n+1\}, A')$. Este grafo é transformado, em GArcos, num grafo orientado, planar e fortemente conexo, $G = (V, A)$. Na última etapa, GDados, alguns dos arcos são escolhidos para formarem o conjunto de arcos de procura (A_R) e são gerados todos os parâmetros necessários à resolução do problema.



Fluxograma 5.1: Geração aleatória de problemas tipo T2

Geração de Grafos Não Orientados, Planares e Conexos

Um grafo associado a uma folha de papel quadriculado, em que os vértices representem as intersecções das quadriculas e as arestas os segmentos delimitadores das quadriculas é planar, não orientado e conexo, sendo de

simples geração. Assim, a obtenção de grafos planares e conexos pode fazer-se a partir da construção de um grafo associado a uma folha de papel quadriculado como se vê na figura seguinte.

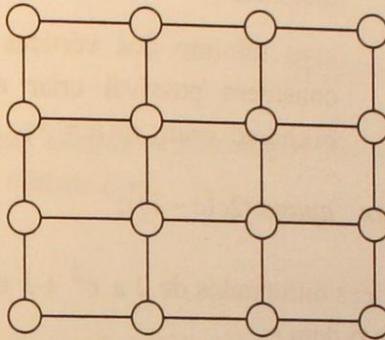


Figura 5.5: Representação do grafo associado a um conjunto de quadriculas

De forma a obter grafos representativos de uma rede de estradas, algumas das arestas iniciais são apagadas, tendo em conta a densidade pretendida e mantendo a conexidade do grafo. A partir de alguns (poucos) dos vértices geram-se também algumas arestas “em diagonal” incidentes nesses vértices, mantendo o grafo resultante planar.

Tendo os vértices 1 e $n+1$ características diferentes das dos restantes vértices do problema, os vértices gerados inicialmente são numerados, sequencialmente, de 2 a $c \times c + 1$, onde c é um valor inteiro, fixo no início. No fim escolhe-se um vértice para representar o vértice 1 , de acordo com o algoritmo seguinte.

Algoritmo GArestas: Geração de um Grafo G' Não Orientado, Planar e Conexo

0. Dados:

- c : sendo $n = c \times c$ o n.º de vértices, com excepção do vértice que representa a ETRS;
- $max_apaga \in [0, 1]$: indicador para o número máximo de arestas a apagar;
- $p_1 \in [0, 1]$: para decidir se uma dada aresta deve ou não ser apagada. Fixou-se $p_1 = 0.3$;
- $n_cruzamax$: número máximo de vértices em que podem incidir arestas “em diagonal”;

- $p_2 \in [0, 1]$: para decidir se deve ou não ser considerado um vértice a partir do qual se criarão arestas “em diagonal”;
- $grau_min$: grau mínimo dos vértices a partir dos quais se considera possível criar arestas “em diagonal”.
Fixou-se $grau_min = 3$;

Fazer: $apaga = \lceil \max_apaga \times 2c(c-1) \rceil$;

1. Considerar os vértices numerados de 2 a $c^2 + 1$ como elementos de uma matriz quadrada de ordem c ;
2. De forma a que o grafo resultante seja representativo de uma folha de papel quadriculado, criar arestas de cada vértice para os vértices
 - 2.1 na mesma linha e na coluna seguinte (caso exista);
 - 2.2 na mesma coluna e na linha seguinte (caso exista);
3. *Repetir* $apaga$ vezes:
 - 3.1 Seleccionar uma aresta, entre as existentes;
 - 3.2 Gerar u , um número pseudo-aleatório (NPA), $U(0, 1)$;
 - 3.3 Se $(u < p_1)$ e (se retirar a aresta não originar um grafo desconexo)
Apagar a aresta seleccionada;
4. Escolher um vértice para representar o Posto de Serviço e considerá-lo como vértice 1;
5. *Repetir* $n_cruzamax$ vezes:
 - 5.1 Gerar u , um NPA, $U(0, 1)$;
 - 5.2 Se $(u < p_2)$

Escolher um vértice, x , diferente do que representa 1, entre os vértices com grau superior ou igual a $grau_min$ ⁽¹⁾;

Repetir para cada uma das arestas “em diagonal” que podem incidir em x e não incidentes em vértices de grau unitário, representativos de “becos”, nem no vértice que representa o vértice 1:

(1) Considera-se possível gerar, a partir deste vértice, arestas “em diagonal”.

Gerar u_2 , um NPA, $U(0,1)$;

Se $(u_2 < p_2)$ e (a inclusão da aresta
mantiver o grafo planar)

Incluir a aresta;

6. Decrementar numa unidade a numeração dos vértices a partir do vértice l ;

Fim.

Exemplo 5.1: Para $c = 3$, resulta, do passo 2.,

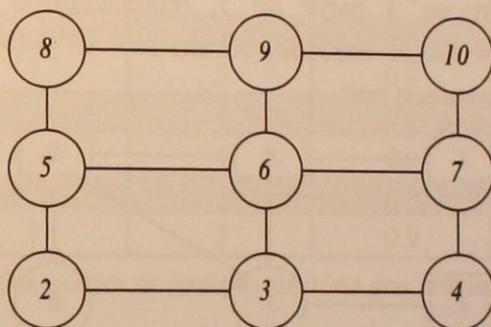


Figura 5.6: Grafo representativo de uma folha quadriculada, para $c = 3$.

3. Suponha-se que, para este grafo, se obtém:

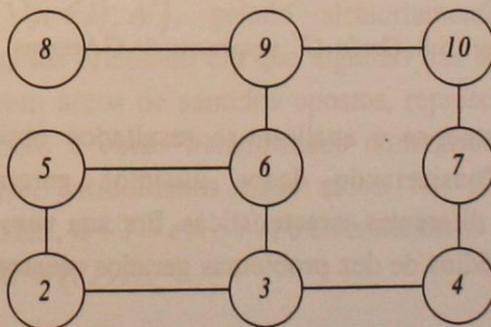


Figura 5.7: Grafo resultante da eliminação de arestas no grafo da figura 5.6.

4. Seja 5 o vértice que representa l ;
5. $n_cruzamax = 2$; $grau_min = 3$

Seja $x = 6$ o primeiro vértice escolhido

As possíveis arestas “em diagonal” a incidir em 6, não sendo consideradas as arestas incidentes em 8 por se tratar de um vértice de grau um, são $(6,2)$, $(6,4)$ e $(6,10)$.

Suponha-se que, em função de u_2 e p_2 , se incluíram as arestas $(6,4)$ e $(6,10)$.

Seja $x = 9$ o segundo vértice escolhido

As arestas "em diagonal" que, eventualmente, poderiam incidir em 9, são $(9,5)$ e $(9,7)$.

Sendo 5 o vértice representativo do vértice 1, $(9,5)$ não é considerada. Também não se considera possível a inclusão de $(9,7)$ por ser incompatível com a existência de $(6,10)$, pois poderia vir a originar um grafo não planar.

6. Todos os vértices, a partir de 5, representativo do vértice 1, são decrementados numa unidade, resultando $G' = (V \setminus \{n+1\}, A')$, que pode ser representado por:

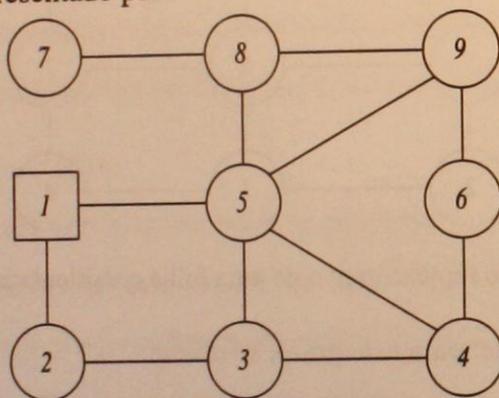


Figura 5.8: Grafo G' resultante de G Arestas

Neste ponto apresentam-se e analisam-se resultados obtidos para diferentes problemas teste. Considerando dados distintos geram-se aleatoriamente problemas teste com diferentes características. Por sua vez, cada problema teste representa valores médios de dez problemas gerados aleatoriamente a partir dos mesmos dados.

Fazendo c variar entre 3 e 7 obtêm-se, como se verá, problemas com 10 a 50 vértices.

Na geração aleatória dos problemas que se apresentarão ao longo deste ponto consideraram-se os valores constantes da tabela 5.1 para os dados de GArestas.

GArestas				
c	Problema n.º	n_cruzamax	p ₂	max_apaga
3	1, 2, 3, 4, 5	2	0.8	0.1
4	1, 2, 3, 4, 5	2	0.8	0.1
5	1,2	2	0.8	0.1
	3	3	0.8	
	4	3	0.9	
	5, 6	4	0.9	
6	1, 2	4	0.8	0.1, 0.2
	3, 4	4	0.9	0.1, 0.2
	5, 6	5	0.8	0.1, 0.2
	7	5	0.9	0.2
7	1, 2	4	0.8	0.1, 0.2
	3, 4	4	0.9	0.1, 0.2
	5, 6	5	0.8	0.1, 0.2
	7	5	0.9	0.2

Tabela 5.1: Dados usados na geração aleatória dos problemas em GArestas

Transformação num Grafo Orientado, Fortemente Conexo e Planar

O grafo $G' = (V \setminus \{n+1\}, A')$, gerado aleatoriamente por GArestas, é transformado num grafo orientado em que algumas das arestas são duplicadas, de forma a originarem arcos de sentidos opostos, representativos de ruas com dois sentidos. O grafo é então transformado num grafo fortemente conexo, sendo, se necessário, introduzidos mais arcos. No fim, cria-se o vértice representativo da ETRS, $n+1 = c \times c + 1$, e consideram-se arcos nele incidentes, obtendo-se o grafo $G = (V, A)$.

Algoritmo GArco: Transformação de G' em G

0. Dados:

- $G' = (V \setminus \{n+1\}, A')$;
- $p_3 \in [0, 1]$: para decidir da duplicação de arestas. Considerou-se $p_3 = 0.5$ ou $p_3 = 0.4$;
- $p_4 \in [0, 1]$: para decidir se um vértice deve ou não ser ligado a $n+1$. Fixou-se $p_4 = 0.2$;

1. Transformar todas as arestas incidentes em vértices de grau um em dois arcos de sentidos opostos, pois estes vértices representam "becos";

2. Para cada uma das restantes arestas, $(i, j) \in A'$, em que nem i nem j representam "becos":

2.1 Gerar u_1 , um NPA $U(0, 1)$;

2.2 Se $(u_1 < p_3)$ Considerar os dois arcos $(i, j) \in A$ e $(j, i) \in A$

C.c., Gerar u_2 , um NPA $U(0, 1)$;

Se $(u_2 < p_3)$ Considerar o arco $(i, j) \in A$

C.c., considerar o arco $(j, i) \in A$;

3. {Transformação do grafo resultante de 2., num grafo fortemente conexo}

Repetir (para todos os vértices $i \in V \setminus \{n+1\}$ que não representem "becos")

Analisar se existe caminho de i para todos os seus possíveis "vizinhos" que não representem "becos", em que se designa por "vizinhos" de um vértice i os vértices $j \in V \setminus \{n+1\}$ que estão na:

- mesma linha e na coluna anterior de i (caso exista);
- mesma linha e na coluna seguinte de i (caso exista);
- mesma coluna e na linha anterior de i (caso exista);
- mesma coluna e na linha seguinte de i (caso exista)

Se não existe caminho de i para j , considerar $(i, j) \in A$;

4. Fazer $n+1 = c^2 + 1$ e criar arcos de/para $n+1$ da seguinte forma:

4.1 Incluir os arcos: $(n+1, c^2)$; $(c^2, n+1)$ (1);

$(n+1, c^2 - 1)$; $(c^2 - 1, n+1)$;

$(n+1, c \times (c-1))$; $(c \times (c-1), n+1)$;

caso nenhum destes vértices represente um "beco";

(1) Esta numeração para os vértices que são sempre unidos a $n+1$ só é válida se o vértice 1 for representado por um vértice de índice inferior a $c \times (c-1)$.

4.2 Fazer $p_5 = \frac{1-p_4}{2}$ (fixou-se $p_4 = 0.2$);

4.3 **Repetir** para cada vértice j que possa ser unido a $n+1$ (1):

Gerar u , um NPA $U(0,1)$;

Se $(u < p_4)$ Criar os arcos $(n+1, j) \in A$ e $(j, n+1) \in A$

C.c., Se $(u < p_4 + p_5)$ Criar o arco $(n+1, j) \in A$

C.c., Criar o arco $(j, n+1) \in A$.

Fim.

Gera-se assim um grafo orientado representando as ruas que podem ser atravessadas em vazio. Este grafo deve ser, obviamente, fortemente conexo.

Exemplo 5.2: Retomando o exemplo 5.1, suponha-se que, dos passos 1. e 2., se obteve

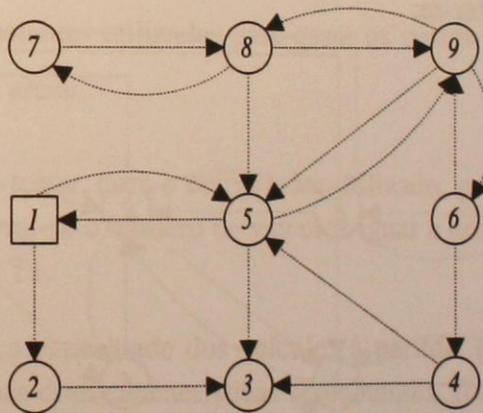


Figura 5.9: Transformação das arestas de G' em arcos de G

O vértice 7, resultando de um vértice de grau unitário de G' , representa um "beco" e, conseqüentemente, consideram-se a incidir neste vértice os dois arcos de sentidos opostos.

(1) Considerou-se que os vértices que podem ser unidos a $n+1$ são os que estão na última linha e na última coluna e ainda não foram considerados.

3. Para os vértices que não representem “becos”, verifica-se a existência de um caminho para cada um dos seus “vizinhos”. Para ilustrar este exemplo apresenta-se a seguinte tabela.

Vértice $i \in V$	“Vizinho” $j \in V$	Caminho de i para j	Criar o Arco
2	3	(2,3)	—
	1	não existe	(2,1)
3	2	Não existe	(3,2)
	4	(3,2,1,5,9,6,4)	—
	5	(3,2,1,5)	—
...

Tabela 5.2: Transformação de G num grafo fortemente conexo

4. No exemplo em análise, os vértices 9, 8 e 6 são unidos a $n+1 = 10$ com dois arcos de sentidos opostos.

Considera-se ainda possível a criação de arcos entre $n+1$ e o vértice 4.

Pode assim obter-se:

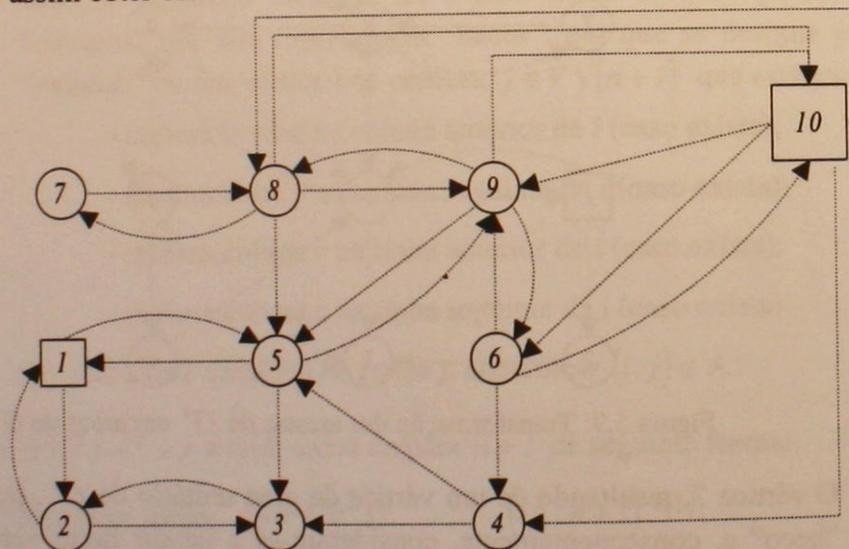


Figura 5.10: Grafo resultante de GArcoS

Geração dos Parâmetros

Gerado o grafo orientado, fortemente conexo e planar, $G = (V, A)$, em GArcoS, pretende-se, neste ponto, explicitar a geração dos parâmetros necessários à resolução do problema. Este procedimento denominou-se por GDados.

Assumem-se conhecidos:

- P : número de veículos. Considerou-se $P = 2$;
- cap : constante inteira utilizada no cálculo da capacidade dos veículos;
- $L1$: constante inteira e positiva representando um valor base no cálculo dos custos de atravessar os arcos em vazio;
- $c_1, c_2 \in (0, 1)$: constantes para a geração dos custos de percorrer os arcos em vazio;
- $p_6 \in [0, 1]$: para fixar arcos de procura entre os arcos de G ;
- $p_7 \in [0, 1]$: para fixar arcos de procura entre os arcos incidentes em "becos";
- $c_4; c_5$: constantes necessárias à geração das procuras associadas aos arcos;
- $c_6; c_7$: constantes utilizadas para gerar os custos de serviço associados aos arcos.

Não sendo possível testar, com o compilador utilizado, problemas com mais de 50 vértices considerou-se o número de veículos igual a dois, tal como tinha sido feito nos problemas T1.

Em vez de se fixar a capacidade dos veículos à partida, por ser assim difícil a geração de problemas com características semelhantes, fixa-se o número médio de subrotas que se pretendem. Assim, dada a constante cap , calcula-se a capacidade dos veículos por:

$$W = \begin{cases} cap & \text{se } cap \geq 100 \\ \left((Q_T + 1) \text{ div } cap \right) \text{ c.c.} & \text{c.c.} \end{cases} \quad (5.2)$$

Se o valor de W calculado por (5.2) não for divisível por dez considera-se para valor da capacidade dos veículos o primeiro múltiplo de dez superior a W .

Assim, para cada um dos problemas teste, foram escolhidos para cap valores inteiros próximos dos valores médios obtidos para $K + P$, constantes das tabelas de resultados computacionais que se apresentam no anexo A3.

Geração dos Custos de Atravessar os Arcos em Vazio

Considerando que u é um NPA, $U(0,1)$, os custos de percorrer, em vazio, cada um dos arcos gerados, são calculados por:

- Arcos não incidentes em $n+1$ e que não sejam “em diagonal”

$$d_{i,j} = \begin{cases} (2u-1) c_1 L1 + L1 & \text{se } d_{j,i} \text{ ainda não foi calculado} \\ (2u-1) c_2 d_{j,i} + d_{j,i} & \text{c.c.} \end{cases} \quad (5.3)$$

Onde $L1$, fixo no início, representa o valor que se estabelece para valor médio dos custos de atravessar os arcos em vazio.

Estes custos foram definidos como sendo valores inteiros, sendo por isso sempre calculados os valores aproximados.

Assim, se $d_{j,i}$ ainda não tiver sido calculado, $d_{i,j}$ assume valores no intervalo $[L1 \mp c_1 L1]$. No caso contrário $d_{i,j} \in [d_{j,i} \mp c_2 d_{j,i}]$.

Sendo $L1$ um valor inteiro positivo, escolhendo $c_1 \in (0;1)$ e $c_2 \in (0;1)$, estes custos assumem, como se pretende, valores positivos.

- Arcos “em diagonal” não incidentes em $n+1$

Defina-se $L2 = \sqrt{2} L1$.

$$d_{i,j} = \begin{cases} (2u-1) c_1 L2 + L2 & \text{se } d_{j,i} \text{ ainda não foi calculado} \\ (2u-1) c_2 d_{j,i} + d_{j,i} & \text{c.c.} \end{cases} \quad (5.4)$$

Se $d_{j,i}$ ainda não tiver sido calculado, $d_{i,j}$ assume então valores no intervalo $[L2 \mp c_1 L2]$.

Assim, $L2$, representando o comprimento da diagonal do quadrado de lado $L1$, representa também o valor médio para o custo dos arcos “em diagonal” que são gerados.

- Arcos incidentes em $n+1$

$$d_{i,j} = \begin{cases} (2u-1)c_1 L + c_3 & \text{se } d_{j,i} \text{ ainda não foi calculado} \\ (2u-1)c_2 d_{j,i} + d_{j,i} & \text{c.c.} \end{cases} \quad (5.5)$$

onde: $c_3 = n(L+1)$.

Neste caso, se $d_{j,i}$ ainda não tiver sido calculado, $d_{i,j}$ assume valores no intervalo $[c_3 \mp c_1 L]$.

Geração dos Arcos de Procura e Respective Parâmetros

Se p_6 , dado no início, for próximo de 1, a maior parte dos arcos gerados, entre os não incidentes em $n+1$, serão considerados arcos de procura.

Um dos dois arcos incidentes em vértices representativos de becos será sempre considerado arco de procura, pois caso contrário não faria sentido considerar esse vértice. Assim, estes arcos são analisados de forma diferente, como se descreve de seguida.

► Escolha dos Arcos de Procura (de A_R)

- Arcos incidentes em vértices que não representem "becos"

Para cada $(i, j) \in A$

Gerar u_1 , um NPA $U(0,1)$;

Se $(u_1 < p_6)$ considerar $(i, j) \in A_R$.

- Arcos incidentes em vértices que representem "becos"

Se $j \in V \setminus \{n+1\}$ representa um "beco" e $(i, j) \in A$ fazer:

$(i, j) \in A_R$;

Gerar u_2 , um NPA $U(0,1)$;

Se $(u_2 < p_7)$ considerar $(j, i) \in A_R$.

Nos problemas teste considerou-se $p_6 = 0.85$ ou $p_6 = 0.75$ e $p_7 = 0.15$.

Assim, embora o grafo G , resultante de G_{Arcos} , represente um grafo fortemente conexo, G_R pode ser formado por diversas componentes conexas.

➤ Geração das Procuras Associadas aos Arcos de A_R

Com o objectivo de relacionar as procuras e os custos de atravessar os arcos em vazio, definiu-se a procura de cada arco por:

$$q_{i,j} = 2 c_5 u + c_4 d_{i,j} - c_5, \quad \forall (i,j) \in A_R. \quad (5.6)$$

Logo, $q_{i,j} \in [c_4 d_{i,j} \mp c_5]$. (5.7)

As constantes fixas no início, devendo proporcionar o cálculo de procuras positivas, podem ser escolhidas verificando:

$$Ll > (1 - c_1)^{-1} \quad \text{e} \quad c_4 > c_5 > 0. \quad (5.8)$$

Pois, $d_{i,j} \geq Ll(1 - c_1) \stackrel{(5.8)}{>} 1 \Rightarrow c_4 d_{i,j} > c_4 \Rightarrow$

$$\Rightarrow c_4 d_{i,j} - c_5 > c_4 - c_5 > 0 \Rightarrow 2 c_5 u + c_4 d_{i,j} - c_5 > 0 \Leftrightarrow q_{i,j} > 0.$$

➤ Geração dos Custos de Serviço Associados aos Arcos de A_R

O custo de serviço associado a cada arco, representando o tempo necessário para efectuar a recolha nesse arco, deve estar de algum modo relacionado com a quantidade de resíduos existente, nunca devendo ser inferior ao custo de percorrer o arco em vazio. Assim, o custo de serviço associado a cada arco $(i,j) \in A_R$, é calculado por:

$$c_{i,j} = 2 c_7 u + c_6 q_{i,j} - c_7, \quad (5.9)$$

assumindo valores em $[c_6 q_{i,j} \mp c_7]$. (5.10)

Se as constantes positivas, c_6 e c_7 , forem tais que

$$c_6 c_4 > 1 \quad \text{e} \quad c_5 c_6 + c_7 < Ll(1 - c_1)(c_6 c_4 - 1) \quad (5.11)$$

resulta, como pretendido, $c_{i,j} > d_{i,j}, \quad \forall (i,j) \in A_R$.

Note-se que se $c_6 c_4 > 1$, então

$$\left. \begin{aligned} d_{i,j} \geq L1(1-c_1) &\Rightarrow d_{i,j}(c_6 c_4 - 1) > L1(1-c_1)(c_6 c_4 - 1) \\ c_5 c_6 + c_7 < L1(1-c_1)(c_6 c_4 - 1) &\Rightarrow \end{aligned} \right\} \Rightarrow$$

$$\Rightarrow d_{i,j}(c_6 c_4 - 1) > c_5 c_6 + c_7 \Leftrightarrow d_{i,j} c_6 c_4 - c_5 c_6 - c_7 > d_{i,j} (*).$$

Por outro lado, por (5.10) e (5.7), vem:

$$c_{i,j} \geq c_6 q_{i,j} - c_7 \geq c_6 (c_4 d_{i,j} - c_5) - c_7 > d_{i,j} (*).$$

Sabendo não ser permitido, no turbo pascal, ter variáveis, declaradas como variáveis inteiras, a assumir valores superiores a "maxint" = 32767, fez-se variar $L1$ entre 15 e 20 e considerou-se $c_1 = 0.1$ ou $c_1 = 0.25$. Para os restantes dados em qualquer dos problemas teste assumiu-se sempre $c_2 = 0.05$, $c_4 = 4$, $c_5 = c_7 = 1$ e $c_6 = 0.5$.

Geram-se assim todos os parâmetros necessários à resolução dos problemas, sendo os valores sempre arredondados a valores inteiros.

5.3.2. *Resultados Computacionais*

Descrita a geração de problemas de tipo T2, vão neste ponto apresentar-se e analisar-se os resultados computacionais.

Como anteriormente frisado, e tal como nos problemas de tipo T1, cada problema teste corresponde a valores médios de dez problemas gerados aleatoriamente com idênticas características.

Nesta tese faz-se um estudo comparativo entre os valores dos desvios relativos proporcionados pelas heurísticas apresentadas, H1, H2 e H3, em problemas teste de tipo T2. Iniciando-se por analisar as heurísticas em separado, estas serão comparadas entre si no final deste capítulo.

Tal como nos problemas de tipo T1, a análise baseia-se nos gráficos que resumem a informação mais relevante, constante das tabelas detalhadas que se apresentam no anexo A3.

Tal como nos problemas anteriores fixou-se em dois o número de veículos.

Resultados Computacionais de H1 em Problemas T2

Os valores percentuais dos desvios relativos da heurística H1 para os problemas de tipo T2 (figura 5.11) podem considerar-se globalmente satisfatórios. Embora não assumam valores inferiores a 14% também nunca ultrapassam os 29%, desvio este que não se pode considerar muito elevado dadas as características do problema em foco.

Embora não seja possível tirar conclusões sobre os valores das soluções em função da variação da capacidade dos veículos ou do número médio de subrotas, parece visível uma certa melhoria dos valores percentuais do "gap" com o aumento do número de vértices. Podem neste caso distinguir-se duas classes de problemas com comportamentos similares. Na classe de problemas com 10 e 17 vértices os valores percentuais do "gap" variam entre 19% e 29%. No respeitante aos problemas com o número de vértices a variar entre 26 e 50 os valores do "gap" situam-se entre 14% e 24%.

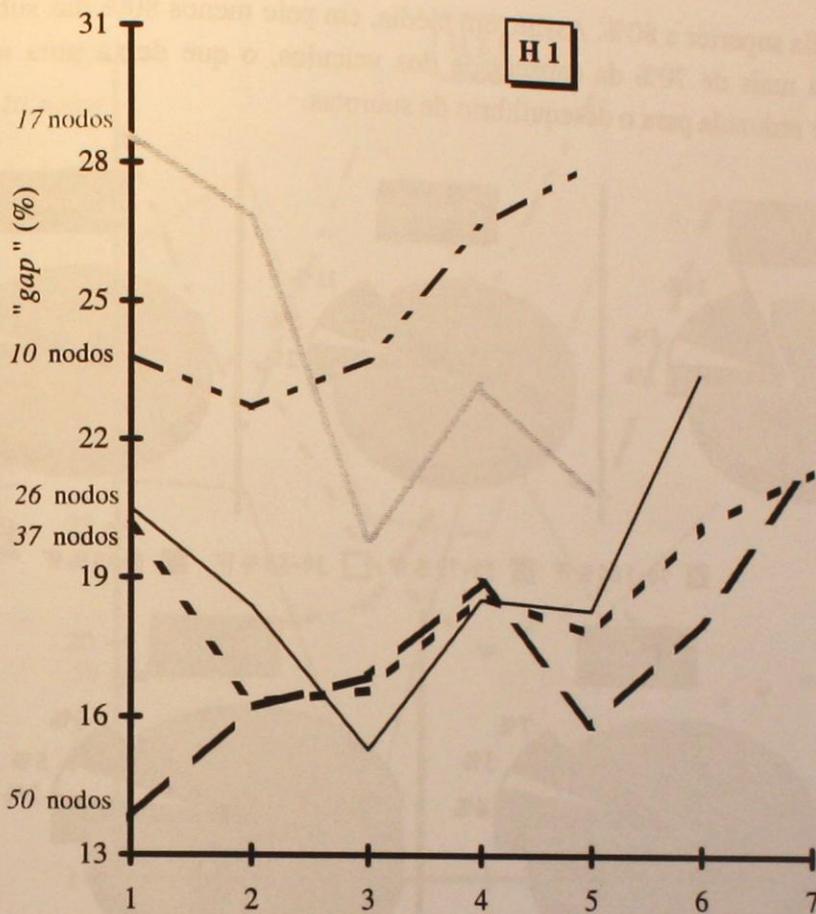


Figura 5.11: Desvios relativos de H1 em problemas T2

O majorante para o pior desvio possível, constante das tabelas em anexo e calculado com base na equação (4.1), assumiu valores médios entre os 90% e os 350%. Como seria expectável, este valor aumenta com o aumento das dimensões do problema. Note-se que o custo do caminho de menor custo mais caro terá tendência a aumentar com o aumento do número de vértices, originando assim um aumento no numerador da equação (4.1).

Tal como anteriormente analisa-se ainda o tipo de subrotas geradas por H1 em problemas T2. Nas tabelas em anexo pode constatar-se que o desvio médio entre o número de subrotas gerado por H1 e $K + P$, ou seja, o número mínimo de subrotas, que se sabe existir em qualquer solução óptima, assume um valor relativamente baixo. Pode assim inferir-se que o número de subrotas gerados por H1 deve estar próximo do número de subrotas de uma solução óptima.

Por outro lado, pode em geral ser considerada razoável a distribuição das procuras pelas diferentes subrotas, informação esta resumida na figura 5.12. Em qualquer dos problemas teste, as subrotas geradas com procura total superior a 70% da capacidade dos veículos nunca assume valores inferiores a 58%, sendo

em média superior a 80%. Assim, em média, em pelo menos 80% das subrotas é ocupada mais de 70% da capacidade dos veículos, o que deixa uma margem bastante reduzida para o desequilíbrio de subrotas.

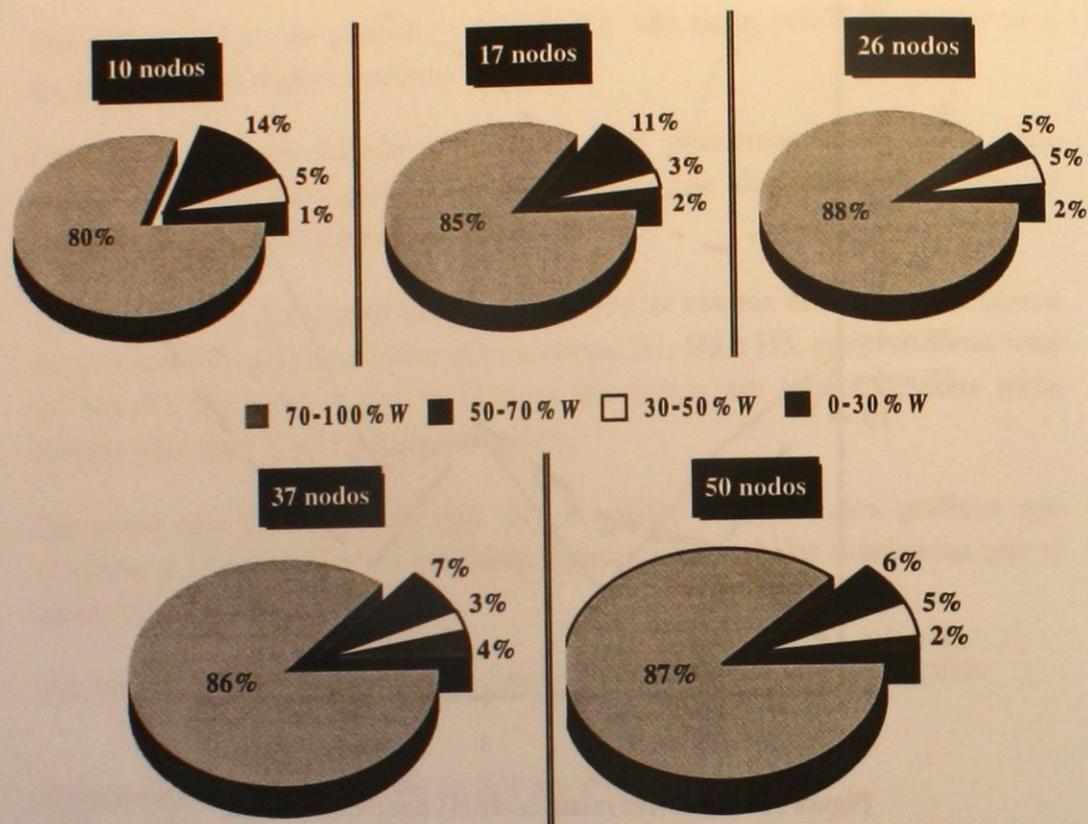


Figura 5.12: Estrutura das subrotas nas soluções admissíveis de H1 em problemas T2

Pode então afirmar-se que os resultados de H1 em problemas de T2 se mostraram “razoáveis”, quer a nível dos desvios relativos do valor das soluções admissíveis relativamente ao valor dos minorantes, quer a nível do tipo de subrotas geradas.

Resultados Computacionais de H2 em Problemas T2

Tal como nos resultados anteriores, os valores percentuais dos desvios relativos da heurística H2 apresentados na figura 5.13 também podem ser considerados satisfatórios. Nota-se neste caso uma variação dos desvios relativos médios entre 15% e 35% para além de uma difícil relação entre os desvios percentuais obtidos e a variação das dimensões dos problemas.

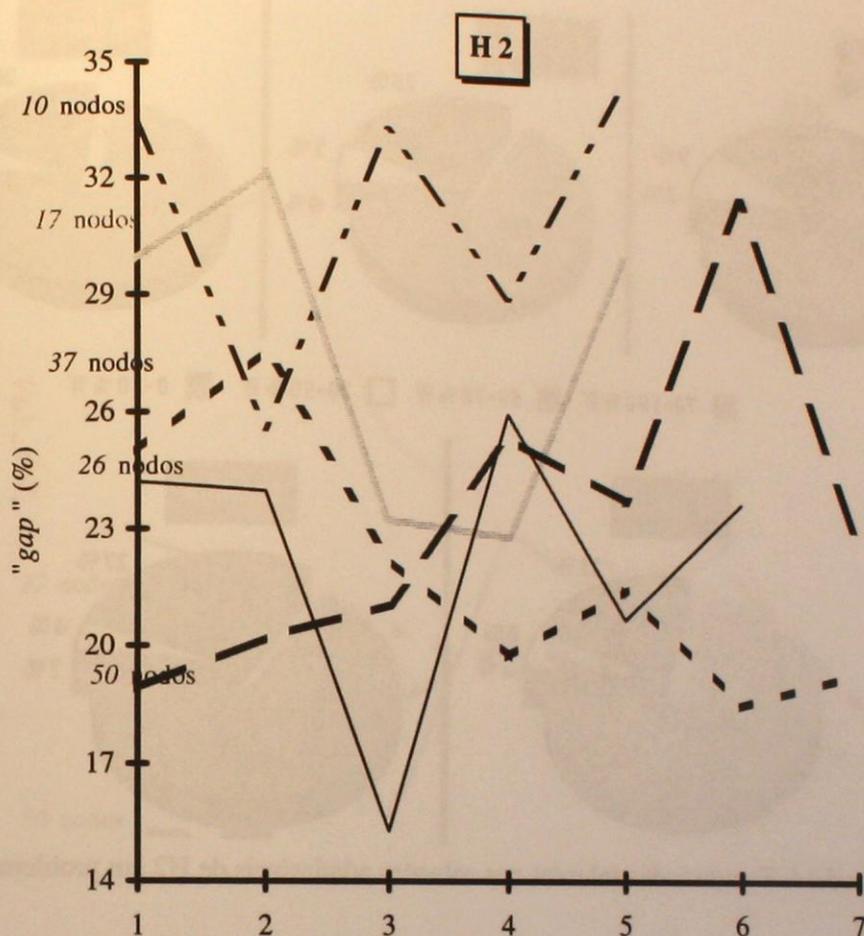


Figura 5.13: Desvios relativos de H2 em problemas T2

Iniciando este método com um conjunto de subrotas admissíveis e sendo em cada iteração agregados diversos pares de subrotas em simultâneo, pensa-se ser normal a geração de um conjunto final de subrotas que não possam ser agregadas, embora com procura total aquém da capacidade dos veículos. Assim se justifica o facto de o desvio entre o número de subrotas geradas por H2 e $K + P$ ser significativo. O aumento do número de subrotas origina, em geral, o aumento no custo das soluções admissíveis e, conseqüentemente, um possível aumento nos valores dos desvios relativos percentuais, ou seja de "gap".

Analisem-se então agora as subrotas geradas por H2 com base nas representações médias da figura 5.14.

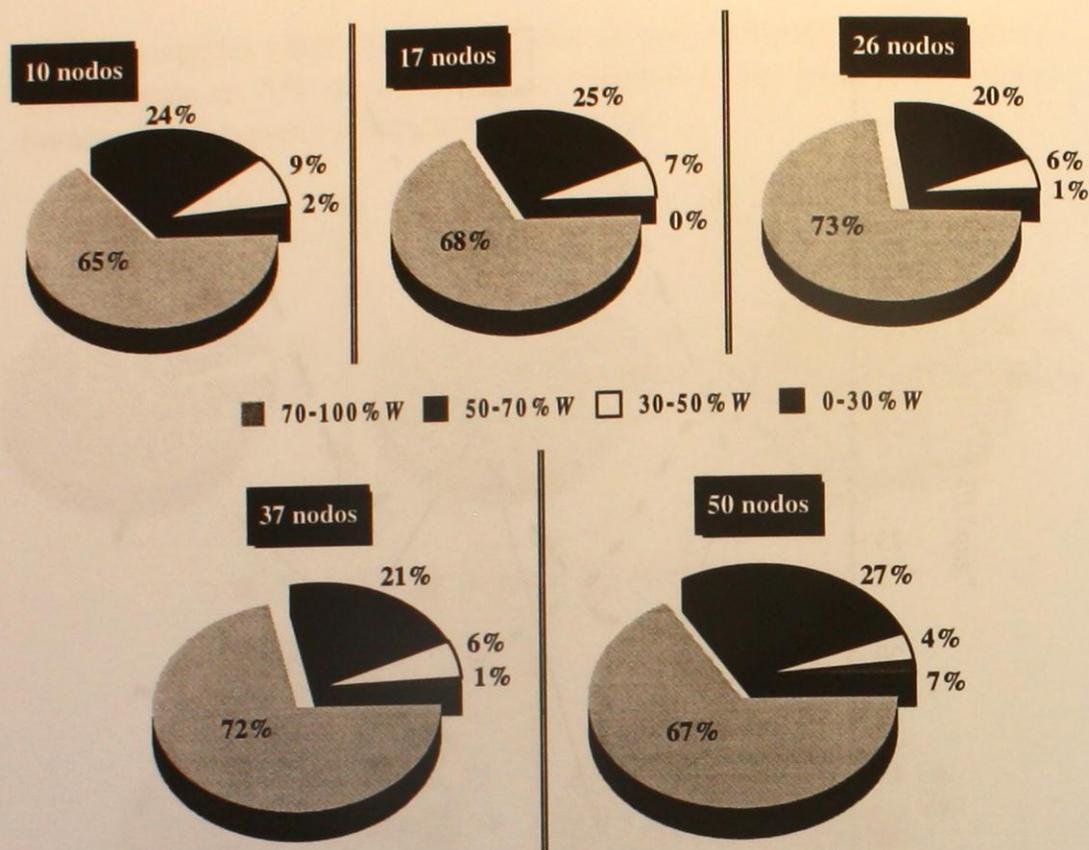


Figura 5.14: Estrutura das subrotas nas soluções admissíveis de H2 em problemas T2

A corroborar o que foi referido na análise dos desvios relativos está o tipo de subrotas obtidas com esta heurística. O número de subrotas em que a procura total varia entre 50 e 70% da capacidade dos veículos é significativo, resultando assim um certo desequilíbrio entre subrotas.

Resultados Computacionais de H3 em Problemas T2

Os desvios percentuais relativos obtidos pela heurística H3 foram, tal como para as restantes heurísticas, resumidos na figura 5.15. Neste caso nota-se uma variação dos valores de "gap" entre 14% e 34%, podendo assim ser considerados valores globalmente satisfatórios.

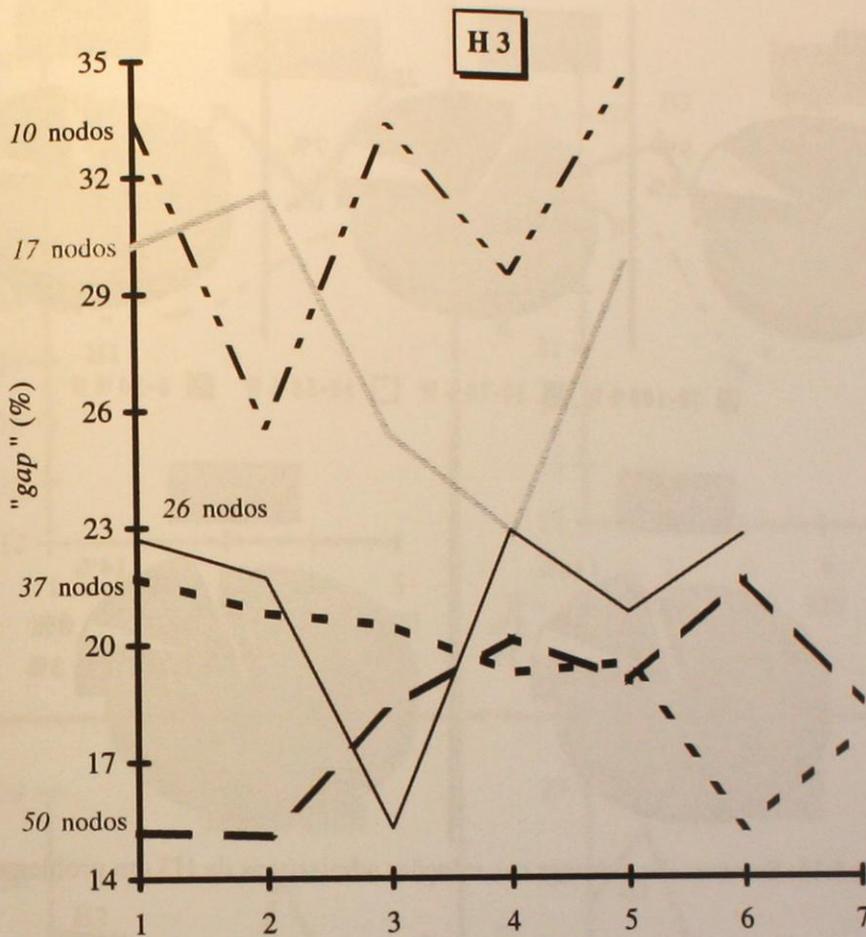


Figura 5.15: Desvios relativos de H3 em problemas T2

Embora com algumas exceções, parece também aqui notória uma certa tendência de melhoria do desvio relativo perante um aumento de dimensões do problema. Iniciando este método, tal como o anterior, com um conjunto de diversas subrotas admissíveis é de esperar que em problemas de pequenas dimensões, como é o caso dos problemas com 10 e com 17 vértices, a fase de agregação não proporcione um conjunto de subrotas muito diferente do inicial. Assim, pode resultar elevado o desvio entre o número de subrotas geradas se for utilizada a heurística H3 e o número de subrotas numa solução óptima, piorando obviamente o valor de "gap".

Note-se que os desvios relativos para problemas com 37 e 50 vértices variam entre 15% e 22%, valores estes significativamente melhores que os encontrados para os problemas com 10 e 17 vértices.

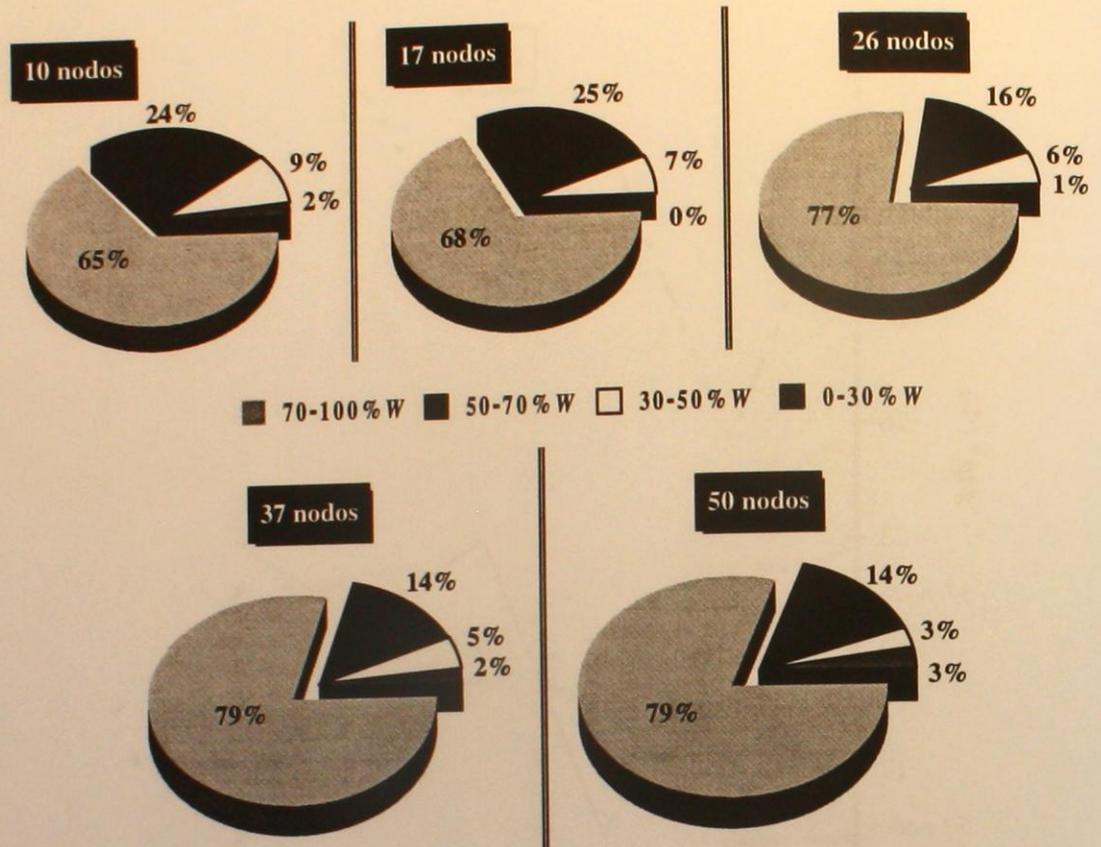


Figura 5.16: Estrutura das subrotas nas soluções admissíveis de H3 em problemas T2

Tal como no respeitante aos valores dos desvios relativos também o conjunto final de subrotas pode ser considerado melhor nos problemas de maiores dimensões. Variando entre 70% e 92% a percentagem de subrotas com capacidade total nunca inferior a 70% da capacidade dos veículos, é forçosamente baixa a percentagem de subrotas em que a procura total não ultrapassa metade da capacidade dos veículos.

Em resumo, este método mostrou-se mais penalizador em problemas de pequenas dimensões. No entanto em maiores dimensões já proporciona soluções admissíveis com valores razoáveis, podendo o conjunto final de subrotas ser considerado satisfatório.

Comparação entre as Heurísticas H1, H2 e H3

Relativamente aos valores obtidos para os desvios relativos com as três heurísticas, H1, H2 e H3, não se pode concluir da dominância de nenhuma delas. Para melhor se compararem os valores obtidos para cada uma das dimensões, desenharam-se os gráficos das figuras 5.17 e 5.18.

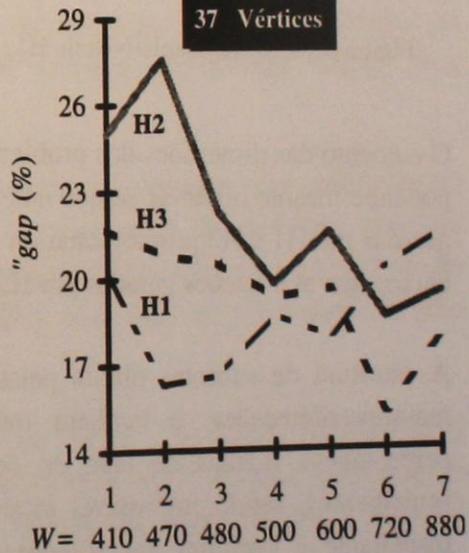
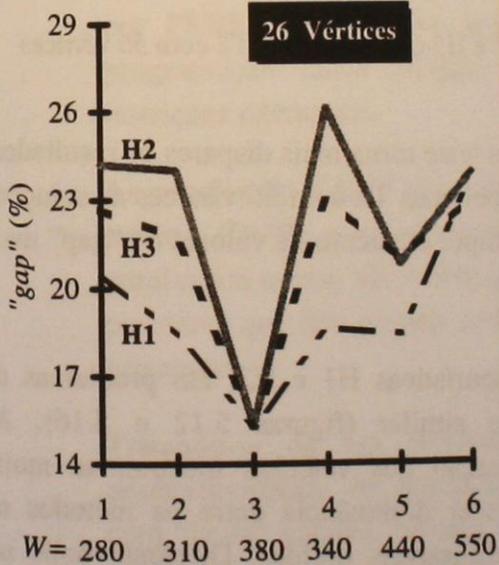
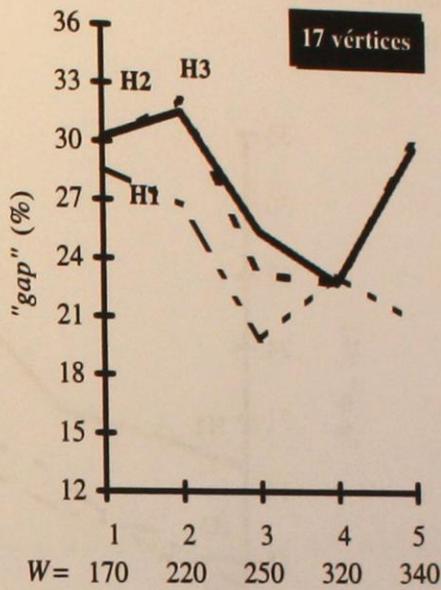
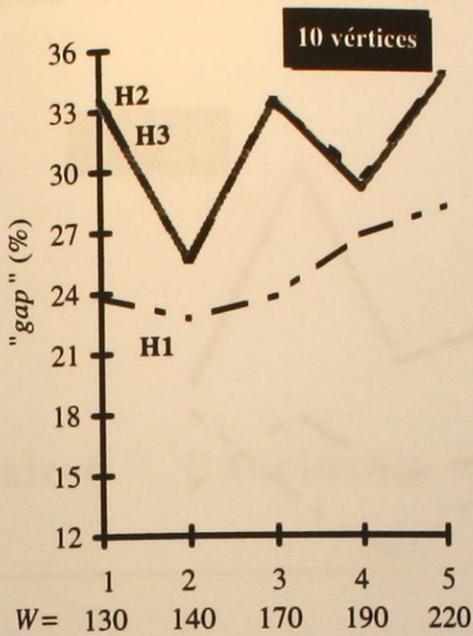


Figura 5.17: Desvios relativos de H1, H2 e H3 em problemas T2 com 10 a 37 vértices

Os valores determinados para os problemas com 10 e 17 vértices poderiam indicar uma certa dominância de H1, proporcionando as heurísticas H2 e H3 soluções muito similares. Como frisado, tal justifica-se pelo facto de nestes métodos ser gerado um conjunto inicial de subrotas que não proporciona muitas alterações na fase de agregação, em problemas de menores dimensões. Sendo a fase de agregação a que distingue as heurísticas H2 e H3, o conjunto final de

subrotas mostrou-se idêntico na maior parte dos problemas teste destas dimensões.

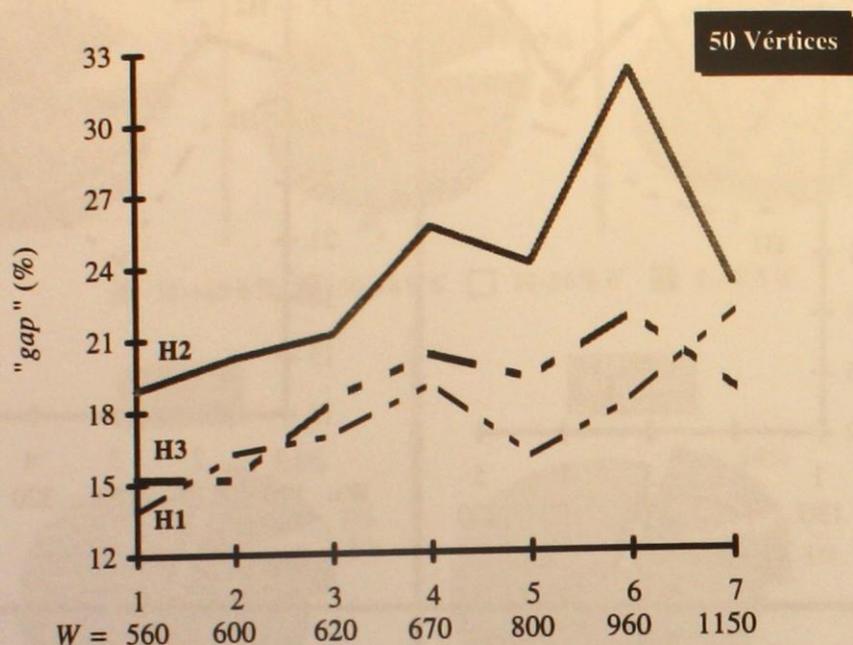


Figura 5.18: Desvios relativos de H1, H2 e H3 em problemas T2 com 50 vértices

O aumento das dimensões dos problemas teste torna mais díspares os resultados, podendo mesmo observar-se que nos problemas T2 com 50 vértices as soluções geradas por H1 e H3 proporcionaram sempre o cálculo de valores de "gap" mais baixos que as soluções geradas por H2.

A estrutura de subrotas obtida pelas heurísticas H1 e H3, em problemas de maiores dimensões, é também muito similar (figuras 5.12 e 5.16). As percentagens médias de taxa de ocupação dos veículos mostram-se muito semelhantes, sendo impossível estabelecer dominância entre os métodos no respeitante à estrutura das soluções admissíveis obtidas. Diferente pode ser considerada a estrutura das subrotas geradas por H2 (figura 5.14).

Em resumo, os valores dos desvios bem como as estruturas das soluções admissíveis proporcionados por ambas as heurísticas H1 e H3 mostraram-se razoáveis em problemas T2. Não se podendo concluir da dominância de nenhum dos métodos, talvez seja de inferir que nos problemas de dimensões pequenas, dado o tipo de subrotas de H3, a heurística H1 é preferível. A heurística H2 proporcionou soluções muito idênticas às de H3 nos problemas de menores dimensões. Face a um aumento nas dimensões dos problemas, H2 foi a heurística que proporcionou a geração de soluções com os maiores valores de "gap" associados.

Capítulo 6. Conclusões

A presente tese versou sobre um problema de optimização de rotas para os veículos afectos à recolha de resíduos sólidos urbanos, designado por PRRS. O problema, após formalizado como um problema de programação linear inteira, foi encarado como um CARP com restrições adicionais.

As versões do CARP mais abordadas na literatura, quer referentes ao próprio CARP, quer se tratem de suas aplicações práticas, abordam geralmente o caso não orientado. Nesta tese, dadas as características do problema que deu origem ao PRRS, enveredou-se pelo estudo do caso orientado.

Tratando-se de um problema de difícil resolução, justifica-se o desenvolvimento de métodos aproximativos que forneçam soluções com "bons" valores, ou seja de valores tão próximos quanto possível do valor óptimo do problema em estudo.

Nesta tese foram deduzidas três diferentes relaxações para o PRRS formalizado. Sendo de minimização o objectivo do PRRS, os valores dos problemas relaxados representam, como se prova, minorantes válidos para o valor óptimo. Mostrou-se ser possível resolver as duas primeiras relaxações por problemas de transporte, enquanto a terceira é resolúvel por um problema de fluxo de custo mínimo.

Na primeira relaxação, como justificado, foram incorporadas restrições de capacidade e de equilíbrio. Na segunda consideraram-se igualmente as restrições de capacidade, parte das restrições de conectividade até então relaxadas e parte das restrições de equilíbrio. Na terceira relaxação consideraram-se obrigatórias todas as restrições de equilíbrio para os vértices equilibrados no grafo de procura, as de capacidade e incorporaram-se também parte das restrições de conectividade. Em consequência, não se conseguiram impor algumas das restrições de equilíbrio nos vértices com grau interno inferior ao externo, no grafo de procura.

Não sendo possível estabelecer uma relação de dominância entre estas relaxações, foram relacionados os seus valores óptimos, para certas classes de instâncias do problema. Foi ainda possível relacionar os valores óptimos dos dois problemas de transporte associados às duas primeiras relaxações para qualquer PRRS.

O algoritmo LB apresentado para obter minorantes para o problema, a partir da primeira relaxação, foi codificado em Pascal, e utilizado em diversos problemas teste para a análise de alguns dos algoritmos descritos nesta tese.

Com o objectivo de obter "boas" soluções admissíveis, ou seja soluções que, sendo admissíveis, tenham custo tão próximo quanto possível do custo óptimo, foram desenvolvidas três heurísticas construtivas. Trata-se de heurísticas de três fases, iniciando, qualquer delas, com a solução associada ao método LB. O multigrafo correspondente à solução deste método, sendo formado por componentes Eulerianas, permite a identificação de circuitos Eulerianos de uma forma simples. Dadas as características deste multigrafo é também possível a sua subdivisão em conjuntos de subcircuitos de procura, pois todo o arco de procura está incluído num circuito. As três fases distintas das heurísticas desenvolvidas consistem em gerar subrotas admissíveis, agregar subrotas e gerar soluções admissíveis. A conjugação de distintos algoritmos para as duas primeiras fases possibilitou a apresentação de três heurísticas diferentes, H1, H2 e H3.

A qualidade das soluções admissíveis obtidas pelas três heurísticas pode ser melhorada com base no método melhorativo descrito. Este método baseia-se em trocas do serviço de alguns arcos entre subrotas diferentes ou na mesma subrota. Assim, caso seja possível e vantajoso, devem substituir-se os arcos percorridos em vazio por arcos percorridos em serviço, tentando evitar muitos percursos em vazio.

Para a primeira heurística, H1, foi possível estabelecer, para cada instância do problema, um majorante para o pior desvio do valor da heurística relativamente

ao valor do minorante. Foi para tal feito um estudo do comportamento da heurística perante uma perspectiva de “pior caso”.

As heurísticas H1, H2 e H3 foram codificadas em Pascal e analisadas as soluções admissíveis que estas proporcionam, quer em termos de custo, quer em termos de composição das próprias soluções.

Para a análise dos métodos implementados foram gerados aleatoriamente dois tipos de problema teste – T1 e T2 – como descrito no capítulo dedicado à apresentação e análise dos resultados computacionais.

Tendo-se obtido resultados muito “bons” para a heurística H1 aplicada aos problemas T1 não foi feita uma análise comparativa de métodos, nesta classe de problemas. As soluções admissíveis proporcionadas por H1 têm valores muito próximos dos valores obtidos para minorantes, por LB, proporcionando valores percentuais para os desvios relativos entre os 0.15% e os 7%, para problemas com 10 a 50 vértices. Sendo nítida uma diminuição dos desvios relativos com o aumento das dimensões dos problemas, pensa-se que para problemas com as características dos problemas T1 estas soluções admissíveis são muito satisfatórias, obtidas num tempo de execução mínimo para as dimensões referidas. Note-se que a qualidade dos desvios relativos está intrinsecamente relacionada com o comportamento quer de H1 quer de LB. Assim, valores “bons” para o “gap” médio indicam que ambos os métodos são “bons”, embora, entre estes só H1 proporcione a obtenção de soluções admissíveis. Associados a estes baixos desvios relativos obtiveram-se ainda soluções constituídas por subrotas equilibradas. Constatou-se que, em média, a procura total em pelo menos 80% das subrotas é superior a 70% da capacidade dos veículos.

Os problemas teste T2, resultantes de grafos esparsos e planares, proporcionaram resultados, em termos dos valores dos desvios relativos, não tão bons como os anteriores, mas que no geral se podem considerar muito razoáveis. A análise de resultados foi também feita com base em problemas de dimensões entre os 10 e os 50 vértices. Como referido, obtiveram-se desvios médios percentuais a variar entre 14% e 29% com a heurística H1, entre 15% e 35% com a heurística H2 e entre 14% e 34% com a heurística H3. Estes valores, tratando-se de desvios relativos, podem ser considerados bastante razoáveis neste tipo de problemas.

Embora esteja patente uma certa dominância da heurística H1 nos problemas de pequenas dimensões, esta em relação a H3 dilui-se perante o aumento da dimensão dos problemas. Com a heurística H2 obtiveram-se em geral soluções admissíveis de custo mais elevado.

Nota-se sem dúvida uma certa necessidade de futuros testes mais exaustivos ao tipo de resultados, quer perante dimensões maiores, quer perante o tipo de parâmetros gerados.

Analisadas foram ainda o tipo de subrotas geradas por cada uma das heurísticas. Assim, foi feito um estudo comparativo entre os subconjuntos de subrotas gerados, no respeitante às procuras totais incluídas nas subrotas quando comparadas com a capacidade dos veículos. Deste estudo pode concluir-se serem similares os conjuntos de subrotas gerados por H1 e H3 em problemas de maiores dimensões. Em média, para pelo menos 80% das subrotas geradas por H1 a procura total incluída é superior a 70% da capacidade dos veículos. No respeitante a H3 este valor é de 65% e de 68% de subrotas para problemas com 10 ou 17 vértices, passando para um valor mínimo de 77% em maiores dimensões. A heurística H3 não deve então ser utilizada na geração de subrotas em problemas de dimensões pequenas, pois fornece, em geral, piores soluções admissíveis que H1.

As subrotas geradas pela heurística H2 têm características distintas das geradas pelas restantes heurísticas, fundamentalmente nos problemas de maiores dimensões. Neste caso, a percentagem de subrotas em que a procura total é superior a 70% da capacidade dos veículos varia entre 67% e 73%. Espera-se poder vir a melhorar os resultados obtidos por esta heurística alterando um pouco a fase de agregação. Uma alternativa em estudo, para problemas de maiores dimensões, consiste em não permitir a agregação inicial de subrotas cuja soma das procuras totais esteja próxima de metade da capacidade dos veículos.

Note-se que um conjunto de rotas equilibradas pode ser obtido a partir de um número de subrotas equilibradas múltiplo do número de veículos. Porém, caso o número de subrotas não seja divisível pelo número de veículos o equilíbrio de subrotas pode traduzir-se num desequilíbrio de rotas, pois pelo menos um dos veículos terá que efectuar mais subrotas que os restantes. Futuramente, pensa-se obter novos resultados computacionais aumentando o número de vértices. Para problemas de maiores dimensões está-se a pensar permitir também um aumento no número de veículos mostrando-se neste caso relevante uma análise ao equilíbrio de subrotas.

Torna-se intuitivo que H2 e H3 exigem maior tempo de execução que H1. Assim, caso haja indiferença de escolha entre as heurísticas para problemas de maiores dimensões, deve ter-se em consideração que H1 domina H2 e H3 relativamente aos tempos de execução necessários.

Os problemas teste que têm surgido na literatura visam apenas o caso não orientado, não servindo portanto para termo de comparação. Esta a razão de não ser feito um estudo comparativo entre os métodos propostos nesta tese e os existentes para o CARP ou problemas similares.

Tendo o PRRS surgido a partir de um problema prático enfrentado pelas câmaras municipais, e embora se tenha enveredado nesta tese por um estudo meramente teórico do problema, pensa-se, se possível, vir a implementar na prática este estudo.

Reeves [80] publicou recentemente um livro dedicado a “novas técnicas heurísticas” no qual incorpora diversas referências bibliográficas relevantes. Ainda neste campo, e tendo em conta os problemas relacionados com o CARP, têm investido na última década fundamentalmente Eglese et al. [29], [30], [32], [58] e Greistorfer [47], [48]. Como seguimento a este trabalho pensa-se ainda tentar abordagens envolvendo estes métodos actualmente em voga.

Of particular note are the efforts of the...

Under the 1911 statute...

It is noted that...

The following...

It is further...

The above...

Anexo A1. Identificação dos Casos Possíveis para o Vértice l

Dadas as restrições do subproblema **(B)** ⁽¹⁾:

$$\left\{ \begin{array}{l} \sum_{(i,l) \in A} f_{i,l} - \sum_{(l,i) \in A} f_{l,i} = D(i) \quad \forall i \in V \end{array} \right. \quad (3.2')$$

$$\left\{ \begin{array}{l} \sum_{(l,i) \in A} f_{l,i} \geq P - d^+(l) \end{array} \right. \quad (3.3')$$

$$\left\{ \begin{array}{l} f_{i,j} \geq 0 \text{ inteiros} \quad \forall (i,j) \in A \end{array} \right. \quad (3.9')$$

pretende identificar-se em que instâncias o vértice l , representando o posto de serviço, pode ser origem e/ou destino. O exemplo seguinte é utilizado para ilustrar os casos que se podem verificar para este vértice.

⁽¹⁾ Capítulo 3., pg. 39.

Exemplo A1.1: Considere-se o subgrafo de procura de um problema com $P = 2$ veículos disponíveis:

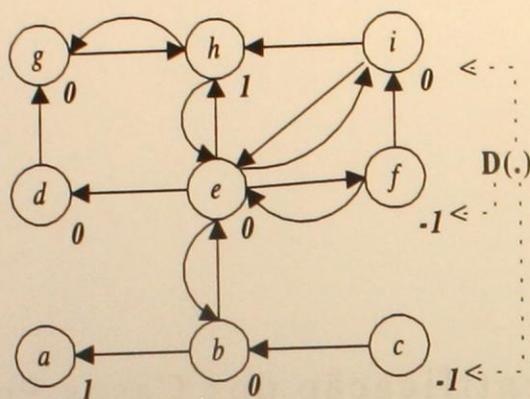


Figura A1.1: grafo de procura

Se:

1. $D(l) = 0$ e

1.a) $P \leq d^+(l) = d^-(l)$

Neste caso l não representa nem origem nem destino, só sendo atravessados em vazio arcos incidentes em l caso seja vantajoso, em termos de objectivo, pois l está equilibrado ($D(l) = 0$) e é possível criar P subrotas iniciais e P subrotas finais se forem apenas considerados os arcos de procura.

Tal como no exemplo A1.1 se o posto de serviço for, por exemplo, o vértice b .

1.b) $P > d^+(l) = d^-(l)$

Por (3.3') o vértice l é origem com oferta $P - d^+(l)$. Como l está equilibrado, o número de vezes que os arcos em vazio com vértice final l são percorridos tem que igualar o número de vezes que são atravessados em vazio arcos com vértice inicial l . Logo, para que (3.2') seja verificada, há que considerar que l representa simultaneamente um destino com procura $P - d^-(l) = P - d^+(l)$.

Se, na figura A1.1, d representar o posto de serviço, d será uma origem e um destino com oferta e procura igual a um, pois:

$$P - d^+(l) = P - d^-(l) = 2 - 1 = 1.$$

2. $D(I) > 0$ e2.a) $P \leq d^-(I)$

Por (3.2') o vértice I é origem com oferta $D(I)$. Ou seja, serão percorridos em vazio pelo menos $D(I)$ arcos com vértice inicial I . Neste caso, as restrições (3.3') ficam automaticamente satisfeitas, pois:

$$\sum f_{I,\ell} \geq D(I) > P - d^+(I).$$

A título exemplificativo considere-se que o vértice h , da figura A1.1, representa I . Obtém-se então $D(I) = 3 - 2 = 1 > 0$ e $P = 2 < 3 = d^-(h)$. Logo I representará uma origem com oferta I . Neste caso podem ser criadas as 2 ($= P$) subrotas iniciais e finais a partir dos arcos de procura incidentes em I , mas é necessário considerar um arco a percorrer em vazio com vértice inicial I , para que este vértice fique equilibrado.

2.b) $P > d^-(I)$

Como $P - d^+(I) > D(I)$, por (3.3') I é origem com oferta $P - d^+(I)$. Neste caso, se considerados apenas os arcos de procura, faltam $P - d^+(I)$ arcos com vértice inicial I para que possam ser geradas P subrotas iniciais. Faltam também $P - d^-(I)$ arcos com vértice final I para que possam ser geradas P subrotas finais. Logo I representará, simultaneamente, um destino com procura $P - d^-(I)$, por imposição de (3.2') e consequência de I ser origem com oferta $P - d^+(I)$.

Considerando que no exemplo o vértice a representa o posto de serviço, vem:

I é uma origem com oferta $P - d^+(I) = 2 - 0 = 2$, para que possam ser criadas duas subrotas iniciais;

I é um destino com procura $P - d^-(I) = 2 - 1 = 1$, para que possam ser criadas duas subrotas finais.

3. $D(I) < 0$ e

3.a) $P \leq d^+(I)$

Por (3.2') o vértice I é destino com procura $-D(I)$. Ou seja, serão percorridos em vazio pelo menos $-D(I)$ arcos com vértice final I . Sendo $P - d^+(I) \leq 0$, as restrições (3.3') são satisfeitas.

Seja I representado pelo vértice f da figura A1.1. Tem-se, $P = 2 = d^+(I)$ e $D(I) = 1 - 2 = -1 < 0$. Logo, I representará um destino com procura I , para que possam ser criadas as 2 subrotas finais. Por outro lado, a partir dos arcos de procura incidentes em I , é possível gerar duas subrotas iniciais ($d^+(I) = 2$).

3.b) $P > d^+(I)$

Este caso é análogo a 2.b). O vértice I deve então ser considerado como uma origem com oferta $P - d^+(I)$ e um como destino com procura $P - d^-(I)$.

Se o vértice c representar o posto de serviço, vem:

$$I \text{ é uma origem com oferta } P - d^+(I) = 2 - 1 = 1;$$

$$I \text{ é um destino com procura } P - d^-(I) = 2 - 0 = 2.$$

O resumo dos resultados anteriores é apresentado na tabela seguinte.

Se		I é	Oferta / procura ($a(I) / b(I)$)
$D(I) = 0$	$P > d^+(I) = d^-(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$
$D(I) > 0$	$P \leq d^-(I)$	origem	$a(I) = D(I)$
	$P > d^-(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$
$D(I) < 0$	$P \leq d^+(I)$	destino	$b(I) = -D(I)$
	$P > d^+(I)$	origem	$a(I) = P - d^+(I)$
		destino	$b(I) = P - d^-(I)$

Tabela A1.1: Casos possíveis para o vértice I relativamente a (PT1)

Anexo A2. Tabelas de Resultados Computacionais para Problemas T1

Neste anexo apresentam-se as tabelas de resultados computacionais para os problemas de tipo T1 gerados aleatoriamente.

O valor do desvio relativo foi calculado pela expressão:

$$gap = \frac{\bar{z} - \underline{z}}{\underline{z}} \times 100, \text{ onde } \bar{z} \text{ representa o valor da solução admissível}$$

encontrada por H1 e \underline{z} o valor do minorante calculado por LB.

Neste tipo de problemas foram geradas redes pouco densas. As procuras foram geradas de acordo com a distribuição uniforme $U(6, 20)$ e os custos associados a cada arco foram gerados pela $U(2, 15)$, para os arcos não incidentes em $n+1$, enquanto para os arcos incidentes em $n+1$ foi utilizada a $U(30, 40)$.

O número de veículos, para esta classe de problemas foi fixo em 2. As capacidades consideradas para cada classe são as indicadas nas tabelas.

Cada linha das tabelas de resultados que se apresentam de seguida referem-se a valores médios de dez problemas gerados aleatoriamente com as mesmas características, ou seja, com o mesmo número de vértices e veículos de igual capacidade.

$n + I$	Problema n.º	W	Q_T	$ A_R $	K + P	H1		% de subrotas que utilizam entre					
						n.º de subrotas	"gap" (%)	0% W e 30% W	30% W e 50% W	50% W e 70% W	70% W e 100% W		
10	1	65	396	33	6.4	7.0	6.5	9	5	3	3	83	
	2	70	530	44	8.1	8.7	6.0	0	9	8	8	83	
	3	80	465	39	6.3	6.6	4.7	1	3	9	9	87	
	4	85	471	38	6.1	6.5	5.5	3	8	9	9	80	
	5	90	520	41	6.3	6.6	4.7	2	6	6	6	86	
Média							5.5	3	6	7	7	84	
15	1	120	1021	85	8.6	9.2	4.1	4	7	4	4	85	
	2	140	1476	108	11.0	11.8	3.5	1	9	4	4	86	
	3	200	1240	100	6.6	7.1	2.9	0	12	18	18	70	
	4	250	1381	118	6.4	6.7	2.3	6	10	7	7	77	
	5	300	1499	120	5.5	5.6	1.2	0	12	7	7	81	
Média							2.8	2	10	8	8	80	
20	1	230	2287	190	10.4	10.8	1.8	0	6	9	9	85	
	2	250	2893	225	12.0	12.8	2.3	3	5	7	7	85	
	3	300	2482	198	8.7	9.1	1.8	4	2	12	12	82	
	4	350	2764	216	8.5	8.9	1.5	4	5	7	7	84	
	5	400	2873	224	7.6	7.9	1.3	2	4	10	10	84	
Média							1.7	3	4	9	9	84	

Tabela A2.1: Resultados computacionais de H1 para problemas de tipo T1 com 10, 15 e 20 vértices

$n + I$	Problema n.º	W	Q_T	$ A_R $	K + P	HI		% de subrotas que utilizam entre				
						n.º de subrotas	"gap" (%)	0% W e 30% W	30% W e 50% W	50% W e 70% W	70% W e 100% W	
25	1	430	3799	299	9.0	9.4	1.2	3	2	8	87	
	2	450	4858	355	11.2	11.9	1.3	2	6	9	83	
	3	500	4085	321	8.7	9.0	1.1	1	6	12	81	
	4	550	4332	340	8.6	8.8	0.9	7	2	6	85	
	5	600	4740	350	8.5	8.6	0.7	0	5	15	80	
Média							1.0	3	4	10	83	
30	1	500	5017	388	10.2	10.4	0.9	2	6	8	84	
	2	600	7153	509	12.2	12.9	1.1	2	5	5	88	
	3	800	6136	462	8.2	8.6	0.8	3	6	17	74	
	4	950	6586	499	8.0	8.1	0.5	2	4	15	80	
	5	1050	7161	502	7.5	7.5	0.5	3	5	10	82	
Média							0.8	2	5	11	82	
35	1	950	7528	594	8.0	8.6	0.7	3	7	7	83	
	2	1050	9986	703	10.0	10.2	0.5	1	2	12	85	
	3	1150	8644	691	7.9	8.4	0.6	2	9	9	80	
	4	1300	9275	699	8.0	8.2	0.4	1	8	12	79	
	5	1400	9788	702	7.5	7.7	0.4	4	5	7	84	
Média							0.5	2	6	9	83	

Tabela A2.2: Resultados computacionais de HI para problemas de tipo T1 com 25, 30 e 35 vértices

$n + l$	Problema n.º	W	Q_T	$ A_R $	K + P	H1		% de subrotas que utilizam entre				
						n.º de subrotas	"gap" (%)	0% W e 30% W	30% W e 50% W	50% W e 70% W	70% W e 100% W	
40	1	950	9915	826	10.6	10.8	0.4	2	2	13	83	
	2	1100	13076	935	12.4	12.7	0.5	1	3	9	87	
	3	1300	11650	914	9.6	9.7	0.4	2	2	12	84	
	4	1450	12417	927	9.2	9.4	0.3	0	2	11	87	
	5	1600	13025	933	8.5	9.2	0.4	1	5	7	87	
Média							0.4	1	3	10	86	
45	1	1200	11470	948	10.0	10.4	0.5	2	4	10	85	
	2	1400	16989	1180	12.7	12.9	0.4	3	3	4	90	
	3	1700	14622	998	9.1	9.5	0.4	3	5	11	81	
	4	1850	15707	1080	9.0	9.5	0.4	1	6	12	81	
	5	2000	14982	1177	8.5	9.0	0.4	8	2	9	81	
Média							0.4	3	4	9	84	
50	1	1500	14972	1244	10.2	11.0	0.4	0	9	11	80	
	2	1700	20536	1456	12.6	12.9	0.4	2	4	6	88	
	3	2100	17323	1269	9.2	9.4	0.2	2	4	6	88	
	4	2300	18753	1362	9.0	9.4	0.3	1	3	9	87	
	5	2500	21173	1466	8.8	9.4	0.3	6	3	5	86	
Média							0.3	2	5	7	86	

Tabela A2.3: Resultados computacionais de H1 para problemas de tipo T1 com 40, 45 e 50 vértices

Anexo A3. Tabelas de Resultados Computacionais para Problemas T2

Neste anexo apresentam-se as tabelas de resultados computacionais para os problemas tipo T2, em redes esparsas e planares, cuja geração se justificou no capítulo 5.

Tal como nos problemas teste de tipo T1, cada problema corresponde a valores médios de dez problemas teste com as mesmas características.

Para cada instância foi calculado um majorante para o erro relativo da heurística H1, constando na tabela, obviamente, os valores médios das dez respectivas instâncias. Relembre-se que a expressão que permite calcular o majorante para o "gap" é:

$$gap \leq \frac{(L + P - 2) lcl(n+1)}{C_{AR} + K scl(n+1)}, \quad (4.1)$$

onde $L = \left\lceil \frac{m}{\ell} \right\rceil$ e $\ell: \sum_{a=1}^{\ell} q_a \leq W < \sum_{a=1}^{\ell+1} q_a$, considerando os arcos ordenados por ordem decrescente de procura.

Nos valores que constam das tabelas seguintes substituiu-se ℓ por

$$\ell' = \left\lceil \frac{W}{q_{max}} \right\rceil, \text{ onde } q_{max} \text{ representa a procura do arco de procura}$$

máxima. Calcula-se assim um majorante para o da equação (4.1).

$n + I$	Problema n.º	W	Q_T	$ A_R $	K + P	N.º de Subrotas			Pior "gap"	"gap" (%)		
						H1	H2	H3		H1	H2	H3
10	1	130	802	13	6.0	6.8	7.0	7.0	103.4	23.8	33.4	33.4
	2	140	961	16	7.0	8.2	8.6	8.6	91.8	22.7	25.6	25.6
	3	170	899	15	5.6	6.8	7.8	7.8	103.4	23.8	33.4	33.4
	4	190	932	15	5.4	6.6	7.0	7.0	116.8	26.8	29.1	29.7
	5	220	861	14	4.0	5.0	5.4	5.4	186.6	28.3	34.8	34.8
17	1	170	1611	27	10.2	12.6	13.2	13.2	137.8	28.6	29.9	30.2
	2	220	1715	28	8.0	9.8	10.6	10.6	132.4	26.8	32.2	31.6
	3	250	1733	29	7.0	8.0	8.6	8.6	164.6	19.9	23.3	25.5
	4	320	1855	30	6.0	7.2	7.4	7.4	204.8	23.3	22.9	22.9
	5	340	1667	27	5.0	5.8	6.6	6.6	202.6	21.0	30.2	30.0
26	1	280	2477	41	9.2	10.4	11.2	11.0	249.6	20.5	24.2	22.7
	2	310	3272	50	10.0	11.2	12.8	12.4	209.0	18.4	24.0	21.7
	3	380	3013	48	8.2	9.4	10.2	10.0	173.8	15.3	15.3	15.3
	4	440	3036	49	7.0	8.0	9.0	8.6	232.8	18.6	26.1	23.0
	5	470	2787	45	6.2	7.0	7.6	7.4	184.0	18.4	20.8	20.9
	6	550	2715	44	5.0	6.0	6.6	6.6	186.4	23.7	23.9	23.0

Tabela A3.1: Resultados computacionais para problemas de tipo T2 com 10, 17 e 26 vértices.

n + I	Problema n.º	H1				H2				H3				
		% de subrotas que ocupam entre												
		0%W a	30%W a	50%W a	70%W a	0%W a	30%W a	50%W a	70%W a	0%W a	30%W a	50%W a	70%W a	100%W
10	1	3	6	33	58	3	6	46	45	3	8	46	43	
	2	4	4	20	72	4	4	44	48	4	4	44	48	
	3	0	3	9	88	3	10	25	62	3	10	25	62	
	4	0	6	6	88	0	8	6	86	0	8	6	86	
	5	0	5	0	95	0	14	0	86	0	14	0	86	
Média		1	5	14	80	2	9	24	65	2	9	24	65	
17	1	0	6	13	81	0	6	24	70	0	8	21	71	
	2	4	0	10	86	0	6	32	62	2	4	30	64	
	3	0	3	8	90	0	2	30	67	0	5	30	66	
	4	0	6	14	81	0	8	16	76	0	8	16	76	
	5	3	0	10	86	0	12	24	64	0	12	27	61	
Média		2	3	11	85	0	7	25	68	0	7	25	68	
26	1	0	8	4	88	0	7	23	70	0	7	16	76	
	2	2	2	2	95	2	6	16	77	0	8	10	82	
	3	4	2	4	89	0	10	14	76	0	8	16	76	
	4	3	3	5	90	4	4	18	73	7	5	12	77	
	5	3	6	6	86	3	0	21	76	0	3	16	81	
	6	3	10	7	80	0	6	27	67	0	6	24	70	
Média		2	3	11	85	0	7	25	68	0	7	25	68	

Tabela A3.2: Composição de subrotas para problemas de tipo T2 com 10, 17 e 26 vértices.

$n + I$	Problema n.º	W	Q_T	$ A_R $	K + P	N.º de Subrotas			Pior "gap"	"gap" (%)		
						H1	H2	H3		H1	H2	H3
37	1	410	4071	66	10.0	11.4	12.8	11.8	286.0	20.1	25.0	21.7
	2	470	4330	70	9.4	10.4	12.6	11.4	306.4	16.3	27.6	20.8
	3	480	4296	70	9.0	10.0	11.2	10.8	270.6	16.6	22.3	20.5
	4	500	3989	66	8.0	9.0	10.2	10.0	191.0	18.7	19.8	19.3
	5	600	4207	69	7.0	8.0	9.0	8.6	288.4	17.9	21.6	19.6
	6	720	4303	72	6.0	7.0	7.4	7.0	180.2	20.3	18.6	15.3
	7	880	4359	71	5.0	6.0	6.6	6.2	198.2	21.6	19.5	17.8
50	1	560	5586	92	10.4	11.2	12.8	11.8	238.4	13.8	18.9	15.2
	2	600	5587	92	9.4	10.4	11.4	10.4	207.0	16.2	20.2	15.1
	3	620	5556	91	9.0	10.0	10.8	10.4	296.6	16.9	21.1	18.4
	4	670	5325	88	8.0	9.0	10.4	9.4	296.0	19.0	25.5	20.2
	5	800	5558	93	7.0	7.8	9.4	8.6	341.8	15.9	23.9	19.1
	6	960	5717	93	6.0	7.0	8.6	7.8	255.8	18.2	31.9	21.8
	7	1150	5714	94	5.0	6.0	6.4	6.2	215.0	21.8	23.0	18.6

Tabela A3.3: Resultados computacionais para problemas de tipo T2 com 37 e 50 vértices.

n + l	Problema n.º	H1						H2						H3					
		% de subrotas que ocupam entre						% de subrotas que ocupam entre						% de subrotas que ocupam entre					
		0%W a	30%W a	50%W a	70%W a	70%W a	100%W	0%W a	30%W a	50%W a	70%W a	70%W a	100%W	0%W a	30%W a	50%W a	70%W a	70%W a	100%W
37	1	0	4	12	84	3	5	23	69	3	5	23	69	3	7	15	75		
	2	2	2	4	92	2	10	29	60	2	10	29	60	5	4	12	79		
	3	2	0	10	88	2	5	14	79	2	5	14	79	4	7	7	81		
	4	0	7	4	89	4	6	12	78	4	6	12	78	4	6	10	80		
	5	3	5	5	88	0	4	18	78	0	4	18	78	0	2	14	84		
	6	9	3	6	83	0	0	24	76	0	0	24	76	0	0	20	80		
	7	10	3	7	80	0	9	27	64	0	9	27	64	0	6	23	71		
Média		4	3	7	86	1	6	21	72	1	6	21	72	2	5	14	79		
50	1	0	0	13	88	3	3	23	70	3	3	23	70	5	0	15	80		
	2	2	6	4	88	2	4	16	79	2	4	16	79	0	4	4	92		
	3	2	4	6	88	0	2	11	87	0	2	11	87	2	4	2	92		
	4	2	2	9	87	0	6	27	67	0	6	27	67	0	6	11	83		
	5	5	3	5	87	2	2	34	62	2	2	34	62	0	0	28	72		
	6	3	11	0	86	2	9	47	42	2	9	47	42	10	10	10	69		
	7	7	7	3	83	3	3	31	63	3	3	31	63	3	0	29	68		
Média		3	5	6	87	2	4	27	67	2	4	27	67	3	3	14	79		

Tabela A3.4: Composição de subrotas para problemas de tipo T2 com 37 e 50 vértices.

Listas

LI. Lista de Algoritmos

Nome	Descrição	Pg.
LB	Determinação de um minorante para o PRRS	51
GSCE	Geração de subrotas admissíveis por partição de circuitos Eulerianos	95
GSProc	Geração de subrotas admissíveis por identificação de circuitos de procura mínima	98
AGSeq	Agregação sequencial de subrotas	107
GS	Construção do multigrafo de subrotas GS	119
AgE	Agregação de subrotas por emparelhamentos	123
AgS	Agregação de subrotas com definição de sementes	128
SolAdm	Geração de uma solução admissível	134
Trocas1	Troca do serviço de arcos numa mesma subrota	145
Trocas2	Troca do serviço de arcos entre subrotas	148
GArestas	Geração de um grafo G' não orientado, planar e conexo	161
GArcos	Transformação de G' em G	165

L2. Lista de Exemplos

N.º	Exemplificação de	Pg.
2.1	Definição de rota, subrota inicial, subrota intermédia e subrota final	21
2.2	Número mínimo de subrotas necessárias	24
2.3	Subrotas não admissíveis	28
3.1	Tipo de circuitos passíveis de serem obtidos com a relaxação 1	38
3.2	Construção do problema (B') a partir do problema (B)	41
3.3	Construção da rede de (PT1) para um caso simples	48
\mathcal{A}	Aplicação do algoritmo LB ao exemplo \mathcal{A}	52
3.4	Existência de componentes conexas equilibradas	55
\mathcal{A}	Construção do grafo \hat{G} para o exemplo \mathcal{A}	61
\mathcal{A}	Construção da rede associada a (PT2) para \mathcal{A}	66
\mathcal{A}	Construção do grafo \tilde{G} para o exemplo \mathcal{A}	74
3.5	Não dominância entre as soluções de (PT1) e de (PT2)	81
\mathcal{A}	Aplicação dos algoritmos GSCE e GSProc ao exemplo \mathcal{A}	101
\mathcal{A}	Aplicação do algoritmo AgSeq ao exemplo \mathcal{A}	110
4.1	Diferenças entre definições de adjacência	116
\mathcal{A}	Aplicação do algoritmo GS ao exemplo \mathcal{A}	121
\mathcal{A}	Aplicação do algoritmo AgE ao exemplo \mathcal{A}	125
4.2	Definição de distâncias em GS	127
\mathcal{A}	Aplicação do algoritmo AgS ao exemplo \mathcal{A}	131
\mathcal{A}	Aplicação do algoritmo SolAdm ao exemplo \mathcal{A}	136
4.3	Trocas de serviço numa subrota	143
4.4	Trocas de serviço entre subrotas	147
\mathcal{A}	Aplicação do método melhorativo ao exemplo \mathcal{A}	151
5.1	Aplicação de GArestas a um exemplo simples	163
5.2	Aplicação de GArcoS a um exemplo simples	167
A1.1	Grafo de procura para ilustrar os casos possíveis para o vértice l em (PT1)	190

L3. *Lista de Figuras*

N.º	Título	Pg.
1.1	Representação de uma rota admissível para um veículo	3
2.1	Pontes de Königsberg	6
2.2	Multigrafo associado às pontes de Königsberg	6
5.1	Desvios relativos de H1 em problemas T1, com 10 a 25 vértices	156
5.2	Desvios relativos de H1 para problemas T1, com 30 a 50 vértices	157
5.3	Estrutura das subrotas nas soluções admissíveis de H1 em problemas T1, com 10 a 25 vértices	158
5.4	Estrutura das subrotas nas soluções admissíveis de H1 em problemas T1, com 30 a 50 vértices	159
5.5	Representação do grafo associado a um conjunto de quadrículas	161
5.11	Desvios relativos de H1 em problemas T2	175
5.12	Estrutura das subrotas nas soluções admissíveis de H1 em problemas T2	176
5.13	Desvios relativos de H2 em problemas T2	177
5.14	Estrutura das subrotas nas soluções admissíveis de H2 em problemas T2	178
5.15	Desvios relativos de H3 em problemas T2	179
5.16	Estrutura das subrotas nas soluções admissíveis de H3 em problemas T2	180
5.17	Desvios relativos de H1, H2 e H3 em problemas T2 com 10 a 37 vértices	181
5.18	Desvios relativos de H1, H2 e H3 em problemas T2 com 50 vértices	182

L4. Lista de Fluxogramas

N.º	Título	Pg.
4.1	Heurísticas	90
4.2	Métodos de agregação por proximidade – AgProxim	114
4.3	Método melhorativo – SAM	143
5.1	Geração aleatória de problemas tipo T2	160

L5. Lista de Tabelas

N.º	Título	Pg.
3.1	Casos que se podem verificar para o vértice 1	40
3.2	Origens de (PT1)	51
3.3	Destinos de (PT1)	52
3.4	Rede para o problema de fluxo de custo mínimo	73
4.1	Sequência de métodos nas diferentes heurísticas	92
5.1	Dados usados na geração aleatória dos problemas em Garestas	165
A1.1	Casos possíveis para o vértice 1 relativamente a (PT1)	192
A2.1	Resultados computacionais de H1 para problemas de tipo T1 com 10, 15 e 20 vértices	194
A2.2	Resultados computacionais de H1 para problemas de tipo T1 com 25, 30 e 35 vértices	195
A2.3	Resultados computacionais de H1 para problemas de tipo T1 com 40, 45 e 50 vértices	196
A3.1	Resultados computacionais para problemas de tipo T2 com 10, 17 e 26 vértices	198
A3.2	Composição de subrotas para problemas de tipo T2 com 10, 17 e 26 vértices.	199
A3.3	Resultados computacionais para problemas de tipo T2 com 37 e 50 vértices	200
A3.4	Composição de subrotas para problemas de tipo T2 com 37 e 50 vértices	201

Referências Bibliográficas

- [1] Álvarez-Valdés, R.; E. Benavent; V. Campos; A. Corberán; E. Mota; J. Tamarit; V. Valls - 1993
"ARC. A Computerized System for Urban Garbage Collection"
Top, 1, pp. 89-105;
- [2] Assad, A.A.; B.L. Golden - 1995
"Arc Routing Methods and Applications"
in: "Network Routing" edited by: M.O. Ball; T.L. Magnanti; C.L. Monma; G.L. Nemhauser; Elsevier Science B.V., North-Holland, Amsterdam;
- [3] Assad, A.A.; W.L. Pearn; B.L. Golden - 1987
"The Capacitated Chinese Postman Problem: Lower Bounds and Solvable Cases"
American Journal of Mathematical and Management Sciences, 7, pp. 63-88;
- [4] Ball, M.O.; M.J. Magazine - 1988
"Sequencing of Insertions in Printed Circuit Board Assembly"
Operations Research, 36, pp. 192-201;
- [5] Ball, M.O.; T.L. Magnanti; C.L. Monma; G.L. Nemhauser - 1995
"Network Routing"
in: "Handbooks in Operations Research and Management Science"; Vol. 8, edited by: G.L. Nemhauser and A.H.G. Rinnooy Kan; Elsevier Science B.V., North-Holland, Amsterdam;
- [6] Beltrami, E.J.; L.D. Bodin - 1974
"Networks and Vehicle Routing for Municipal Waste Collection"
Networks, 4, pp. 65-94;

- [7] Benavent, E.; V. Campos; A. Corberán; E. Mota - 1990
"The Capacitated Arc Routing Problem. A Heuristic Algorithm"
Qüestiió, 14, pp. 107-122;
- [8] Benavent, E.; V. Campos; A. Corberán; E. Mota - 1992
"The Capacitated Arc Routing Problem: Lower Bounds"
Networks, 22, pp. 669-690;
- [9] Bodin, L.D. - 1975
"A Taxonomic Structure for Vehicle Routing and Scheduling Problems"
Computers and Urban Society, 1, pp. 11-29;
- [10] Bodin, L.; B. Golden - 1981
"Classification in Vehicle Routing and Scheduling"
Networks, 11, pp. 97-108;
- [11] Bodin, L.D.; S.J. Kursh - 1978
"A Computer-Assisted System for the Routing and Scheduling of Street Sweepers"
Operations Research, 26, pp. 525-537;
- [12] Bodin, L.D.; S.J. Kursh - 1979
"A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweepers"
Computers and Operations Research, 6, pp. 181-198;
- [13] Bodin, L.; L. Levy - 1991
"The Arc Partitioning Problem"
European Journal of Operational Research, 53, pp. 393-401;
- [14] Bodin, L.; G. Fagin; R. Welebny; J. Greenberg - 1989
"The Design of a Computerized Sanitation Vehicle Routing and Scheduling System for the Town of Oyster Bay, New York"
Computers and Operations Research, 16, pp. 45-54;
- [15] Bodin, L.D.; B.L. Golden; A.A. Assad; M. Ball - 1983
"Routing and Scheduling of Vehicles and Crews: A State of the Art"
Computers and Operations Research, 10, pp. 63-211;
- [16] Caruso, C.; A. Colomi; M. Paruccini - 1993
"The Regional Urban Solid Waste Management System: A Modelling Approach"
European Journal of Operational Research, 70, pp. 16-30;
- [17] Chapleau, L.; J.A. Ferland; G. Lapalme; J.M. Rousseau - 1984
"A Parallel Insert Method for the Capacitated Arc Routing Problem"
Operations Research Letters, 3, pp. 95-99;
- [18] Christofides, N. - 1973
"The Optimum Traversal of a Graph"
Omega, 1, pp. 719-732;

- [19] Christofides, N.; V. Campos; A. Corberán; E. Mota - 1981
"An Algorithm for the Rural Postman Problem"
Imperial College Report IC.OR.81.5, London, England;
- [20] Christofides, N.; V. Campos; A. Corberán; E. Mota - 1982
"An Algorithm for the Rural Postman Problem on a Directed Graph"
Imperial College Report IC.OR.82.18, London, England;
- [21] Clark, R.M.; J.I. Gillean - 1975
"Analyses of Solid Waste Management Operations in Cleveland, Ohio: A Case Study"
Interfaces, 6, pp. 32-42;
- [22] Clark, R.M.; J.C.H. Lee - 1976
"Systems Planning for Solid Waste Collection"
Computers and Operations Research, 3, pp. 157-173;
- [23] Clarke, G.; J.W. Wright - 1964
"Scheduling of Vehicles from a central Depot to a Number of Delivery Points"
Operations Research, 12, pp. 568-581;
- [24] Cook, T.M.; B.S. Alprin - 1976
"Snow and Ice Removal in an Urban Environment"
Management Science, 23, pp. 227-234;
- [25] Corberán, A.; J.M. Sanchis - 1994
"A Polyhedral Approach to the Rural Postman Problem"
European Journal of Operational Research, 79, pp. 95-114;
- [26] Desrochers, M.; J.K. Lenstra; M.W.P. Savelsbergh - 1990
"A Classification Scheme for Vehicle Routing and Scheduling Problems"
European Journal of Operational Research, 46, pp. 322-332;
- [27] Dror, M.; H. Stern; P. Trudeau - 1987
"Postman Tour on a Graph with Precedence Relation on Arcs"
Networks, 17, pp. 283-294;
- [28] Edmonds, J.; E.L. Johnson - 1973
"Matching, Euler Tours and the Chinese Postman"
Mathematical Programming, 5, pp. 88-124;
- [29] Eglese, R.W. - 1990
"Simulated Annealing: A tool for Operational Research"
European Journal of Operational Research, 46, pp. 271-281;
- [30] Eglese, R.W. - 1994
"Routeing Winter Gritting Vehicles"
Discrete Applied Mathematics, 48, pp. 231-244;

- [31] Eglese, R.W.; L.Y.O. Li - 1992
"Efficient Routeing for Winter Gritting"
Journal of Operational Research Society, 43, pp. 1031-1034;
- [32] Eglese, R.W.; L.Y.O. Li - 1996
"LOCSAR: A Local Search Algorithm for Constrained Arc Routing Problems"
Presented at IFORS96, Vancouver, Canada;
- [33] Eglese, R.W.; H. Murdock - 1991
"Routeing Road Sweepers in a Rural Area"
Journal of Operational Research Society, 42, pp. 281-288;
- [34] Eiselt, H.A.; M. Gendreau; G. Laporte - 1995
"Arc Routing Problems, Part I: The Chinese Postman Problem"
Operations Research, 43, pp. 231-242;
- [35] Eiselt, H.A.; M. Gendreau; G. Laporte - 1995
"Arc Routing Problems. Part II: The Rural Postman Problem"
Operations Research, 43, pp. 399-414;
- [36] Euler, L. - 1953
"Leonard Euler and the Königsberg Bridges"
Edited by: J.R. Newman, Scientific American, 189, pp. 66-70;
- [37] Evans, J.R.; E. Minieka - 1992
"Optimization Algorithms for Networks and Graphs"
Second Edition, Marcel Dekker, Inc., New York;
- [38] Floyd, R.W. - 1962
"Algorithm 97, Shortest Path"
Communication ACM, 5, pp. 345;
- [39] Ford, L.R.; D.R. Fulkerson - 1974
"Flows in Networks"
Princeton University Press, Sixth Printing;
- [40] Frederickson, G.N. - 1979
"Approximation Algorithms for some Postman Problems "
Journal of the ACM, 26, pp. 538-554;
- [41] Frederickson, G.N.; M.S. Hetch; C.E. Kim - 1978
"Approximation Algorithms for some Routing Problems"
SIAM Journal on Computing, 7, pp. 178-193;
- [42] Golden, B.L.; R.T. Wong - 1981
"Capacitated Arc Routing Problems"
Networks, 11, pp. 305-315;

- [43] Golden, B.L.; J.S. DeArmon; E.K. Baker - 1981
"Algorithms for the Capacitated Chinese Postman Problem"
Working Paper MS/S #81-024, College of Business and Management, University of Maryland at College Park;
- [44] Golden, B.L.; J.S. DeArmon; E.K. Baker - 1983
"Computational Experiments with Algorithms for a Class of Routing Problems"
Computers and Operations Research, 10, pp. 47-59;
- [45] Gondran, M.; Minoux, M. - 1984
"Graphs and Algorithms"
John Wiley & Sons;
- [46] Gottinger, H.W. - 1988
"A Computational Model for Solid Waste Management with Application"
European Journal of Operational Research, 35, pp. 350-364;
- [47] Greistorfer, P. - 1994
"Computational Experiments with Heuristics for a Capacitated Arc Routing Problem"
Working Paper, N. 32, Department of Business, University of Graz, Austria;
- [48] Greistorfer, P. - 1994
"Algorithms and Implementations for the Mixed Capacitated Chinese Postman Problem"
Working Paper, N. 33, Department of Business, University of Graz, Austria;
- [49] Guan, M. (Kwan Mei-Ko) - 1962
"Graphic Programming Using Odd or Even Points"
Chinese Mathematics, 1, pp. 273-277;
- [50] Guan, M. (Kwan Mei-Ko) - 1962
"Improvement on Graphic Programming"
Chinese Mathematics, 1, pp. 278-287;
- [51] Hirabayashi, R.; Y. Saruwatari; N. Nishida - 1990
"Tour Construction Algorithm for the Capacitated Arc Routing Problem"
Technical Report Number 26, Science University of Tokyo, Tokyo 162, Japan;
- [52] Jansen, K. - 1993
"Bounds for the General Capacitated Routing Problem"
Networks, 23, pp. 165-173;
- [53] Lenstra, J.K.; A.H.G. Rinnooy Kan - 1976
"On General Routing Problems"
Networks, 6, pp. 273-280;
- [54] Lenstra, J.K.; A.H.G. Rinnooy Kan - 1981
"Complexity of Vehicle Routing and Scheduling Problems"
Networks, 11, pp. 221-227;

- [55] Levy, L.; L. Bodin - 1988
 "Scheduling the Postal Carriers for the United States Postal Service: An Application of Arc Partitioning and Routing"
 in: B. L. Golden and A. A. Assad (eds.); "Vehicle Routing: Methods and Studies"; pp. 359-394, Elsevier Science Publishers B.V., North-Holland, Amsterdam;
- [56] Levy, L.; L. Bodin - 1989
 "The Arc Oriented Location Routing Problem"
 INFOR, 27, pp. 74-94;
- [57] Li, L.Y.O.; R.W. Eglese - 1992
 "A Lower Bound for the Constrained Arc Routeing Problem"
 Presented at Combinatorial Optimization, CO92; Oxford, United Kingdom;
- [58] Li, L.Y.O.; R.W. Eglese - 1994
 "A Comparison of Simulated Annealing and Tabu Search for a Constrained Arc Routing Problem"
 Unpublished manuscript;
- [59] Lin, S. - 1965
 "Computer Solutions of the Traveling Salesman Problem"
 Bell System Technical Journal, 44, pp. 2245-2269;
- [60] Lin, S.; B. Kernighan - 1973
 "An Effective Heuristic Algorithm for the Traveling Salesman Problem"
 Operations Research, 21, pp. 498-516;
- [61] Lin, Y.; Y. Zhao - 1988
 "A New Algorithm for the Directed Chinese Postman Problem"
 Computers and Operations Research, 15, pp. 577-584;
- [62] Male, J.W.; J.C. Liebman - 1978
 "Districting and Routing for Solid Waste Collection"
 Journal of the Environmental Engineering Division, 104 (EE1), pp. 1-14;
- [63] Male, J.W.; J.C. Liebman; C.S. Orloff - 1977
 "An Improvement of Orloff's General Routing Problem"
 Networks, 7, pp. 89-92;
- [64] Marks, D.H.; J.C. Liebman - 1970
 "Mathematical Analysis of Solid Waste Collection"
 U.S. Department of Health, Education and Welfare, Public Health Service Publication No. 2104, Bureau of Solid Waste Management;
- [65] McBride, R. - 1982
 "Controlling Left and U-turns in the Routing of Refuse Collection Vehicles"
 Computers and Operations Research, 9, pp. 145-152;

- [66] Minieka, E. - 1979
"The Chinese Postman Problem for Mixed Networks"
Management Science, 25, pp. 643-648;
- [67] Mourão, M.C.; M.T. Almeida - 1993
"Optimização de Rotas para Veículos com Capacidade Limitada em Problemas com Procura nos Arcos: Minorante"
CEMAPRE, Documento de Trabalho n.º 7-93;
- [68] Newton, R.M.; W.H. Thomas - 1974
"Bus-Routing in a Multi-School System"
Computers and Operations Research, 1, pp. 213-222;
- [69] Or, I. - 1990
"Improving the Efficiency of the Solid Waste Collection and Transportation System in IZMIR and Network Techniques"
Presented at IFORS 90, Athens, Greece;
- [70] Orloff, C.S. - 1974
"A fundamental Problem in Vehicle Routing"
Networks, 4, pp. 35-64;
- [71] Orloff, C.S. - 1974
"Routing a Fleet of M Vehicles to/from a Central Facility"
Networks, 4, pp. 147-162;
- [72] Orloff, C.S. - 1976
"Route Constrained Fleet Scheduling"
Transportation Science, 10, pp. 149-168;
- [73] Papadimitriou, C. - 1976
"On the Complexity of Edge Traversing"
Journal of Association on Computing Machinery, 23, pp. 544-554;
- [74] Pearn, W.L. - 1988
"New Lower Bounds for the Capacitated Arc Routing Problem"
Networks, 18, pp. 181-191;
- [75] Pearn, W.L. - 1989
"Approximate Solutions for the Capacitated Arc Routing Problem"
Computers and Operations Research, 16, pp. 589-600;
- [76] Pearn, W.L. - 1991
"Augment-Insert Algorithms for the Capacitated Arc Routing Problem"
Computers and Operations Research, 18, pp. 189-198;
- [77] Pearn, W.L. - 1994
"Solvable Cases of the k -Person Chinese Postman Problem"
Operations Research Letters, 16, pp. 241-244;

- [78] Pearn, W.L.; A. Assad; B.L. Golden - 1987
"Transforming Arc Routing into Node Routing Problems"
Computers and Operations Research, 14, pp. 285-288;
- [79] Ralphs, T.K. - 1993
"On the Mixed Chinese Postman Problem"
Operations Research Letters, 14, pp. 123-127;
- [80] Reeves, C.R. - 1993
"Modern Heuristic Techniques for Combinatorial Problems"
Blackwell Scientific Publications, Oxford;
- [81] Riccio, L.J.; A. Litke - 1986
"Making a Clean Sweep: Simulating the Effects of Illegally Parked Cars on New York
City's Mechanical Street-Cleaning Efforts"
Operations Research, 34, pp. 661-666;
- [82] Roy, S.; J.M. Rousseau - 1988
"The Capacitated Canadian Postman Problem"
Publication #587, Centre de Recherche sur les Transports, Université de Montréal,
Canada;
- [83] Russell, R.; W. Igo - 1979
"An Assignment Routing Problem"
Networks, 9, pp. 1-17;
- [84] Saruwatari, Y.; R. Hirabayashi; N. Nishida - 1991
"Node Duplication Lower Bounds for the Capacitated Arc Routing Problem"
Technical Report, Science University of Tokyo, Tokyo 162, Japan;
- [85] Saruwatari, Y.; R. Hirabayashi; N. Nishida - 1991
"Subtour Elimination Algorithm for the Capacitated Arc Routing Problem"
Technical Report, Science University of Tokyo, Tokyo 162, Japan;
- [86] Shuster, K.A.; D.A. Schur - 1974
"Heuristic Routing for Solid Waste Collection Vehicles"
US Environmental Protection Agency, SW-113; U.S. Government Printing Office,
Washington, D.C.,
- [87] Stern, H.I.; M. Dror - 1979
"Routing Electric Meter Readers"
Computers and Operations Research, 6, pp. 209-223;
- [88] Syslo, M.; N. Deo; J.S. Kowalik - 1983
"Discrete Optimization Algorithms with Pascal Programs"
Prentice Hall, Inc.;

- [89] Turner, W.C.; E.S. Houglan - 1975
"The Optimal Routing of Solid Waste Collection Vehicles"
AIIE Transactions, 7, pp. 427-431;
- [90] Ulusoy, G. - 1985
"The Fleet Size and Mix Problem for Capacitated Arc Routing"
European Journal of Operational Research, 22, pp. 329-337;
- [91] Wunderlich, J.; M. Collette; L. Levy; L.Bodin - 1992
"Scheduling Meter Readers for Southern California Gas Company"
Interfaces, 22, pp. 22-30;
- [92] Zaw Win - 1988
"Contributions to Routing Problems"
PhD. Dissertation, Universität Augsburg.